REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-98-

0824

ring
1 of
uite

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December, 1995	3. REPORT Final

4. TITLE AND SUBTITLE
USAF Summer Research Program - 1995 Graduate Student Research Program Final Reports, Volume 9, Rome Laboratory

5. FUNDING NUMBERS

6. AUTHORS
Gary Moore

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Research and Development Labs, Culver City, CA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NI
4040 Fairfax Dr, Suite 500
Arlington, VA 22203-1613

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Contract Number: F49620-93-C-0063

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The United States Air Force Graduate Student Research Program (USAF- GSRP) is designed to introduce university, college, and technical institute graduate students to Air Force research. This is accomplished by the graduate students being selected on a nationally advertised competitive basis during the summer intersession period to perform research at Air Force Research Laboratory Technical Directorates and Air Force Air Logistics Centers. Each participant provided a report of their research, and these reports are consolidated into this annual report.

14. SUBJECT TERMS
AIR FORCE RESEARCH, AIR FORCE, ENGINEERING, LABORATORIES, REPORTS, SUMMER, UNIVERSITIES

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

UNITED STATES AIR FORCE

SUMMER RESEARCH PROGRAM -- 1995

GRADUATE STUDENT RESEARCH PROGRAM FINAL REPORTS

VOLUME 9

ROME LABORATORY

RESEARCH & DEVELOPMENT LABORATORIES

5800 Uplander Way

Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Major David Hart

Program Manager, RDL
Scott Licoscos

Program Administrator, RDL
Gwendolyn Smith

Program Administrator, RDL
Johnetta Thompson

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

December 1995

19981214 073

PREFACE

Reports in this volume are numbered consecutively beginning with number 1. Each report is paginated with the report number followed by consecutive page numbers, e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3.

This document is one of a set of 16 volumes describing the 1995 AFOSR Summer Research Program. The following volumes comprise the set:

VOLUME	TITLE
1.	Program Management Report
	Summer Faculty Research Program (SFRP) Reports
2A & 2B	Armstrong Laboratory
3A & 3B	Phillips Laboratory
4	Rome Laboratory
5A, 5B, & 5C	Wright Laboratory
6A & 6B	Arnold Engineering Development Center, Wilford Hall Medical Center and Air Logistics Centers
	Graduate Student Research Program (GSRP) Reports
7A & 7B	Armstrong Laboratory
8	Phillips Laboratory
9	Rome Laboratory
10A & 10B	Wright Laboratory
11	Arnold Engineering Development Center, Wilford Hall Medical Center and Air Logistics Centers
	High School Apprenticeship Program (HSAP) Reports
12A & 12B	Armstrong Laboratory
13	Phillips Laboratory
14	Rome Laboratory
15A&15B	Wright Laboratory
16	Arnold Engineering Development Center

GSRP FINAL REPORT TABLE OF CONTENTS   i-xiv

GSRP FINAL REPORTS

AMBIGUITY AND CONFLICT
IN
INFORMATION SYSTEMS DEVELOPMENT -
A CASE STUDY OF WEATHER DATA INTEGRATION

Raymond J. Barnes
PhD Candidate
Dept. of Industrial Engineering and Systems Science

Thomas J. Watson School of Engineering
Binghamton University
P.O. Box 6000
Binghamton, NY 13902

AMBIGUITY AND CONFLICT
IN
INFORMATION SYSTEMS DEVELOPMENT -
A CASE STUDY OF WEATHER DATA INTEGRATION

Raymond J. Barnes
PhD Candidate
Dept. of Industrial Engineering and Systems Science
Binghamton University

Abstract

An analysis of ambiguity and conflict in Information Systems Development (ISD) is performed through a case study involving specification of functional requirements for a common Weather Toolkit (WxTK). The WxTK will allow Air Force geographic information system applications to automate the integration of on-line digital weather data. Given the potentially diverse range of application needs the WxTK would serve, ambiguity inherent in function specifications may obscure important issues, leading to conflicts during later phases of development. In defining the WxTK architecture and library functions the research addresses current ambiguities about what the system must do, and identifies potential conflicts involving the design and implementation of the WxTK. Importantly, ambiguity and conflict per se are not deemed as "good" or "bad", but rather inherent components of systems development in an environment of rapid technological change and volatile user needs. By recognizing the significance of clarifying development issues at requirements specification time, unanticipated conflict may be minimized during the design (building) and implementation phases. The research provides the foundations of a comprehensive framework for dealing with requirements ambiguity and conflict.

AMBIGUITY AND CONFLICT
IN
INFORMATION SYSTEMS DEVELOPMENT -
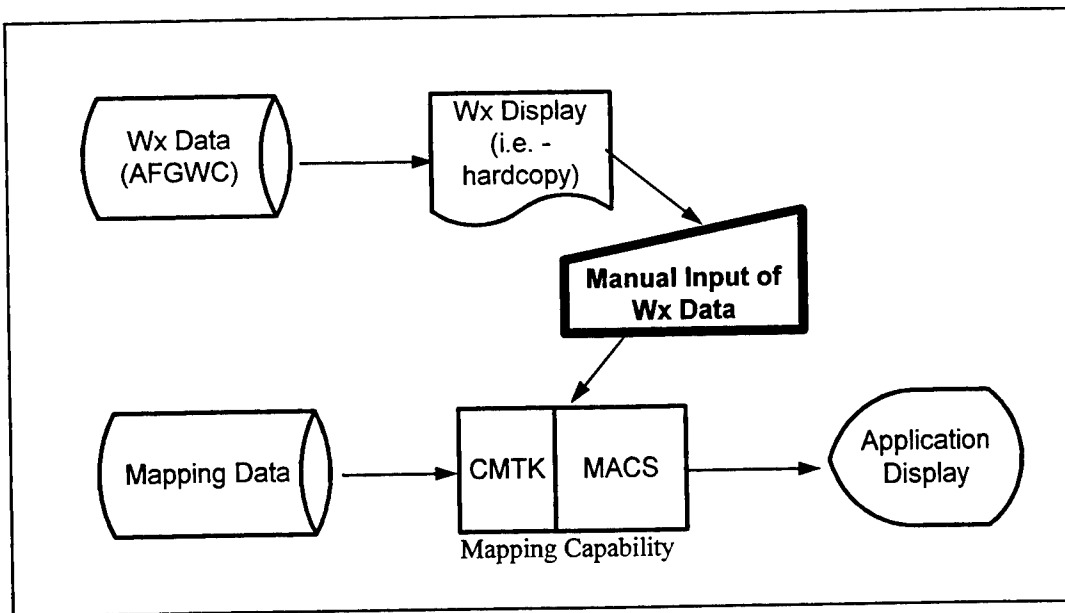## A CASE STUDY OF WEATHER DATA INTEGRATION

Raymond J. Barnes

1. <u>Introduction</u>

1.1 Background - The Need for Automated Integration of Weather Data with Air Force Geographic Information Systems (GIS)

A conspicuously absent functionality of United States Air Force Geographic Information Systems (GIS) is automated integration of on-line weather data. Although an abundance of digital weather data products are available, GIS applications such as Rapid Application of Air Power (RAAP), Advanced Planning System (APS), and Force Level Execution (FLEX) currently enter weather information manually into their map database or display, rather than directly integrating digitized weather data. Recognizing the need to automate on-line weather data integration with Air Force mapping systems, a small-scale (10,000 LOC) rapid prototype of a Weather Tool (WxTool) was developed for Rome Laboratory by Sterling Software. The WxTool demonstrates functionality for access, query and display of on-line weather information using the Common Mapping Toolkit/Mapping Applications Client-Server (CMTK/MACS) {1} as the underlying mapping capability. However, the WxTool prototype is only a static model of what could be done; it does not provide application programmers with a clear definition of functions for integrating weather data, nor access to those functions.

Figure 1 portrays the gap between Air Force mapping applications and the Air Force Global Weather Central (AFGWC) on-line database. An opportunity exists for applications to access an abundance of digitized, on-line weather data products and automatically integrate data for display or as inputs to decision models. For example, RAAP requires data such as temperature and visibility in supporting mission planning. Currently weather data are manually integrated, typically through a customized weather data input screen as read from a separate weather data source (hardcopy report, screen, etc.). Given the multitude of data available, manual integration tends to be labor intensive and error prone, providing relatively little incentive for exploiting

Figure 1
Current Method of Wx Data Integration
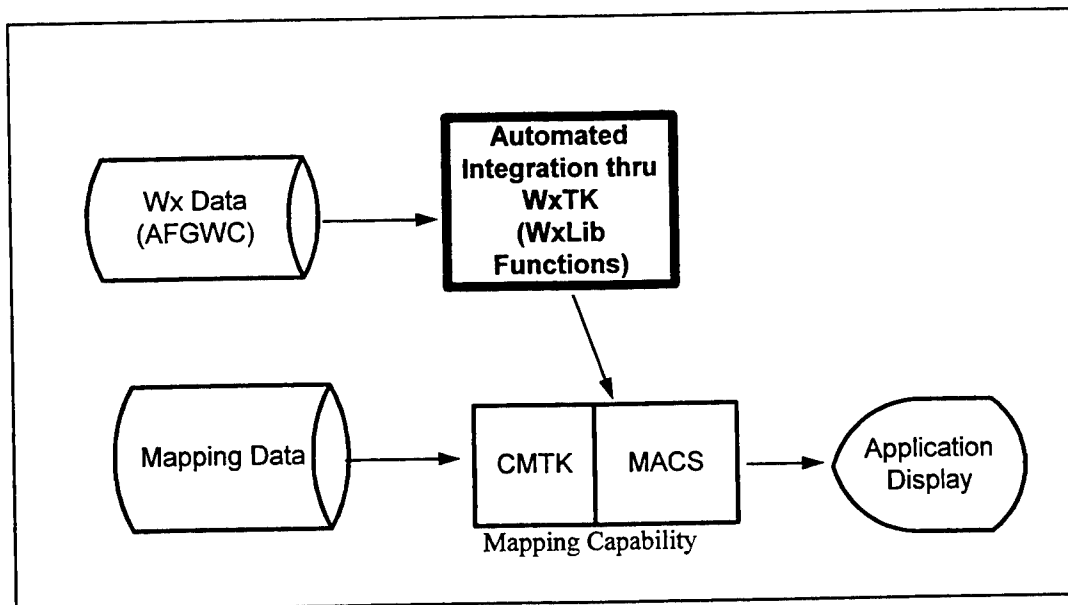
Figure 2
Automated Data Integration with the WxTK

weather data. Figure 2 shows how the weather data integration gap is filled by the WxTK, primarily through a library of C functions known as Wx Library (WxLib) functions. The WxLib functions provide the means for reading and manipulating AFGWC on-line weather data (specifically gridded data products) by GIS applications. The research is focussed on defining and documenting a preliminary set of functional requirements and software architecture for a Weather ToolKit (WxTK), based on the WxTool prototype. The defined WxTK functions are embodied as the WxLib functions, which are available to application programmers. The WxLib functions serve as the foundation for developing a comprehensive WxTK, and can be used as a common set of "tools" to build automated weather data integration by a wide range of GIS applications.

1.2 Analysis of Ambiguity and Conflict in WxLib Function Specifications

Information systems design literature has suggested that ambiguity and conflict in requirements specifications are among the leading challenges currently faced by developers ({2}, {3}, {4}). Since the WxLib functions serve a wide array of user needs, an understanding of ambiguities and conflicts involving functional requirements (what the system must do) is crucial to the success of the WxTK in becoming a truly common tool. An analysis of ambiguity and conflict is incorporated into the development of functional requirements for automated weather data integration. The main research goal involves identifying significant ambiguities about requirements issues which entail potential conflicts involving the current WxLib functional specifications.

A major thrust of the research hinges on the questions of "Why are ambiguity and conflict not recognized in ISD (or why do they persist throughout development), and why does it matter?". One answer may lie in the philosophical underpinnings of ISD. Researchers have pointed to the strong influence of functionalist and rationalist philosophies on traditional ISD practice {2}. Functionalism is associated with an objectivist epistemology of a "one best way" of knowing and understanding the world. From a systems development point of view, functionalism implies each part of the system can be specified unambiguosly and uniquely identified with its function. Rationalism has also significantly influenced engineering and management science {5}, and subsequently software engineering and ISD methodology. A rationalist ISD view contends developers can enumerate all alternative solutions and choose a uniquely optimal or "best" solution to system development problems, or at least identify sub-optimal but satisfactory

solutions ("bounded rationality"). Functionalist and rationalist philosophical positions tend to view conflict over definitions of the world (and systems) as errors or simply a lack of information, rather than as equally valid yet opposing positions.

The efficacy of functionalist/rationalist-based approaches to ISD is questionable when dealing with the rapidly developing technologies and diverse user requirements faced by information system developers today. Designers possessing a high degree of technical knowledge and experience may tend to underestimate the potential for ambiguity and conflict in functional specifications, assuming they "know what users require", without the necessity of any further clarification or debate. Others may simply be overwhelmed by the complexity of the system environment or the pressures of schedule and budget to press the clarification of requirements issues. The research will show the clarification of functional requirements issues and identification of potential conflicts to be a non-trivial task, which has yet to be explored and described in-depth by the ISD and software development literature.

Precise definitions of qualitative concepts like ambiguity and conflict can be difficult to derive, and open to multiple interpretation. Definitions are therefore offered for each term as they are used. Sources for the following definitions include sociological {6} and management science {7} literature and standard dictionary definitions {8}:

Ambiguity: A vagueness, indefiniteness or uncertainty about system requirements, perhaps based on a lack of knowledge, understanding and comprehension or communication of users' needs.

Issue: A point, matter or question (involving system requirements) to be decided, that is potentially contentious.

Conflict: A protracted fight or struggle between parties in a development project over inconsistent/incompatible goals or wants/needs for what the system must do (struggle over issues). Struggles require consumption of resources to resolve (time, money and attention).

No claim is made that unnecessary ambiguity and conflict can be avoided, due to the philosophical difficulties involved in defining what "unnecessary" means. Instead, a model is proposed suggesting ambiguity inherent in the world (origin of requirements) can be passed
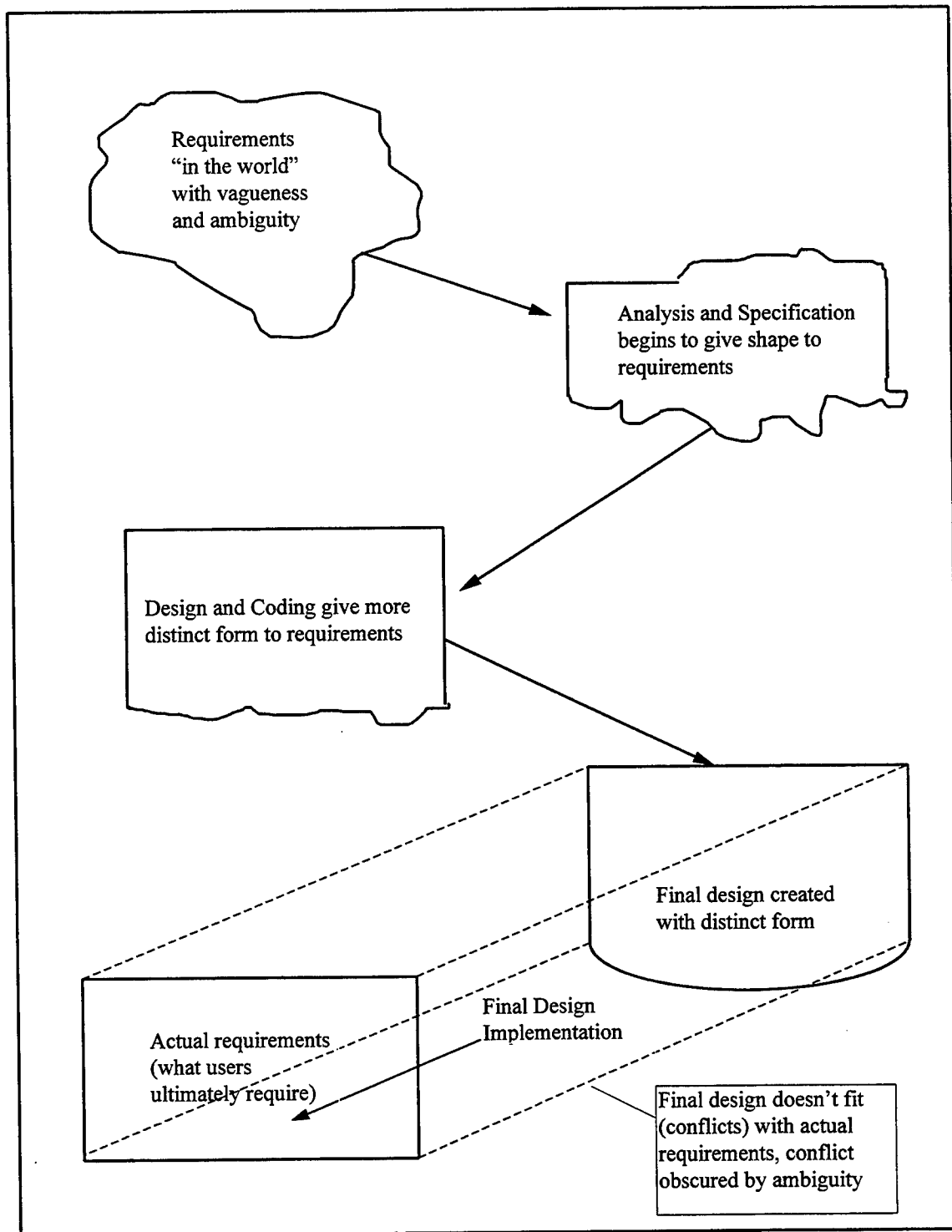
Figure 3
Model of Ambiguity and Conflict in ISD

along through requirements specification, obscuring important issues (Figure 3). Ambiguity in requirements specification can then be passed to the design stage, shrouding design issues. Struggles may then emerge as designers interpret ambiguous requirements in potentially

2.   Research Goals

2.1   Extract useful functions from the WxTool prototype and design a "Weather Application Program Interface" library (WxLib). The WxLib functions will be specified in a generic, stand-alone format where possible, allowing a wide range of applications to integrate AFGWC weather data. The WxLib will form the foundation of the WxTK.

2.2   Demonstrate the WxLib functions through a prototype application (WxTest) using the C programming language with X-Windows and Motif to build the interface. The demonstration will focus on potential Theater Battle Management (TBM) core system modules which use weather data such as RAAP, APS and FLEX applications, as well as weather experts.

2.3   Suggest additional functional requirements for a Weather Toolkit not present in the WxTool and WxTest Application for future (Phase II) development.

2.4   Evaluate the preliminary WxLib functional requirements and describe ambiguity about issues and the resulting potential for conflict. The goal is to identify issues for clarification and resolve conflicts at requirements specification time (if feasible), rather than dealing with conflicts during the design phase, or allowing them to be built into the product.

3.   Research Methodology

A case study approach is used in analyzing the role of ambiguity and conflict in development of the WxLib functions in order to get in-depth, "hands on" experience with the task of functional requirements specification. The case study methodology allows the researcher to look behind the issues and probe into questions of "how" and "why" observed phenomena take place {10}. Case study results do not imply generalized inferences about a phenomena, and no statistically significant conclusions about the general population of ISD or Air Force projects can be drawn. Instead, a case study allows a detailed probe of a real-life functional requirements specification in order to discover how a particular development effort encounters ambiguity and conflict. The

results will provide the conceptual foundation for more broad-based studies of ambiguity and conflict in systems development, and suggest how ISD philosophy may need to be reconsidered.

4.   The WxLib Development Approach

Developing the WxLib functions has essentially followed the Reverse Engineering {11} and Rapid Prototyping {12} approaches. The WxTool was created to quickly demonstrate how an application may combine digital on-line weather data with a mapping capability such as the CMTK/MACS. WxTool documentation is not available, only the prototype source code and an executable file exist. Development of the WxLib functions and definition of the WxTK software architecture involved evaluating the WxTool source code functions and running the executable file to identify basic functionality.

A "ambiguity and conflict" perspective was taken during definition of the WxLib functions, by consciously identifying issues about functions needing clarification, which may lead to potential conflict if not resolved. No formal guidelines for identifying ambiguity and conflict were followed, only the definitions described in section 2. The goal is not to simply specify a monolithic set of WxLib functions that imply inherent unity and coherence, but also describe associated ambiguities involving function design, and how those ambiguities may lead to conflicts in the future. Though conflict may not be avoided, potential struggles can be tracked and monitored throughout the development process rather than ignored or forgotten, only to unexpectedly emerge later and delay progress.

5.   AFGWC On-Line Weather Data Products

An abundance of digital weather data products are available from AFGWC in a variety of formats. AFGWC on-line products are updated periodically depending on availability and frequency of observations. The current WxLib functions use only uniform gridded data format products. However, future development of the WxTK may include all the AFGWC on-line data products. The weather data listed below (and the procedure for obtaining them) are described in detail in the AFGWC Dial-in Subsystem (AFDIS) Software Users Manual {13}.

5.1   Textual Products: Textual data products include weather observations at specific locations, forecasts and bulletins. Textual products are human-readable files in ASCII

format, though the information is coded (for example, "OFF" for Offutt Air Force Base, etc.). The current WxLib functions do not read or manipulate textual data products.

5.2   Vector Graphic Products: Graphical information such as wind barbs and other weather symbology can be included directly in weather data files as vector objects, at specified screen display coordinates corresponding to latitude/longitude. The current WxLib functions do not read vector data products, due to unresolved problems with overlaying display of vector objects with a specific display capability used by an application (i.e. the Spatial Display Tool of the CMTK/MACS).

5.3   Raster Graphic Products: Raster products include scanned-in satellite photos, maps and charts depicting weather information. The WxTool prototype has no functionality for displaying weather raster graphics products, and neither do the WxLib functions. The CMTK/MACS Spatial Display Tool can display raster satellite imagery (i.e. ADRI), and therefore the WxTK could eventually incorporate raster imagery display capability.

5.4   Uniform Gridded Data Products: Uniform Gridded Data products are the only file types supported by the current WxLib functions. Gridded products represent numeric values for weather data (such as temperature) placed at uniformly-spaced intervals resulting in a grid, such as a 2.5 x 2.5 degree latitude/longitude grid {14}. Gridded data do not represent actual observations at specified points, but estimates based on computerized models calculated by AFGWC. The grids are defined by a particular geospatial scheme called the Projection Indicator (PI) Set, where each grid point is assigned an "I" and "J" coordinate. The I-J coordinate values can be associated with a latitude/longitude, but the corresponding values must be calculated. In other words, a given I-J coordinate must be converted to the associated latitude/longitude in order to be displayed with a mapping capability such as the CMTK/MACS, and a latitude/longitude value must be converted to I-J format in order to search a given grid. Gridded data can be obtained for different levels of resolution depending on availability and size of the area of interest.

6.   Functional Requirements for the Weather Library (WxLib)
WxLib functions were specified primarily from assessing the capability of the WxTool rapid prototype (by Reverse Engineering), and by interviewing application developers and weather

experts. The resulting categories of library functions are defined below. The WxLib functions were defined in a generic format when possible, where the only system requirements are that the C programming and SUN/Unix environments be used. Not all the WxLib functions are generic, particularly those which involve display of weather data.

A detailed description of the WxLib functions can be found in the WxTest Application prototype available from Rome Laboratory - IRRP, along with source code and executable file. The reader should note this is not a detailed or comprehensive listing of all WxLib functions found in the WxTool. Other potentially useful functions in the WxTool have yet to be documented and included in the WxLib (see Undeveloped functions).

6.1   Weather Database Read (WxDBRead)

WxDBRead functions open files and read file header information for file type, date, area of coverage, etc.. Functions were defined to access weather data downloaded from AFGWC using the Air Force Dial-in System software (AFDIS), and only gridded data products are assumed to be read. A description of the principal data structure for the gridded data file header format ("dirnames") can be found in the header file "Wx.h" (included with the WxTest package). A utility function is also defined called "WxMakeList()" which takes a Motif list widget and creates a list of weather data products for the application user. WxMakeList() requires the use of X-Windows and Motif by the application, and is intended as an example of a time saving "rapid-prototyping" function.

6.2   Load Data (WxLoad)

The WxLoad functions load gridded weather data values in memory, reading file grid information (number of points, mesh level, value at the I-J point, etc.) and building necessary data structures in memory. The principal data structure for the gridded data values is a substructure of "dirnames" called "sgdb_header" (where sgdb is "Satellite Global Data Base"). Definition of the sgdb_header data structure is also defined in the header file Wx.h.

6.3   Building and Displaying Iso-lines (WxContour)

WxContour functions define algorithms for building iso-lines (lines connecting points of equal values, such as temperature) which can then be displayed. The contouring algorithms are standard routines which are not dependent on any particular mapping capability or hardware

configuration. However, functions for displaying iso-lines are dependent on the CMTK/MACS Spatial Display Tool API's.

6.4   Coordinate Conversions (WxCrdCnv)

Numerous functions for performing coordinate conversions are found in the WxTool rapid prototype. Two functions incorporated in the WxTest application prototype are described here. Converting latitude/longitude coordinates to I-J values (ConvertLatLonToIJCoords()) is useful in performing point queries, where the user enters a specific latitude/longitude and the system looks up the weather data value at the corresponding I-J point in the grid (which is loaded in memory by WxLoadFile()). Conversion of I-J coordinate points (ConvertIJToLatLonCoords()) is necessary for displaying AFGWC data with a mapping display such as the CMTK/MACS, where the corresponding latitude/longitude positions are required. Other subroutines not listed here perform coordinate conversions based on hemisphere location and for display of other file types (satellite visual and infrared imagery).

6.5   Point Data Query (WxQuery)

The WxQuery functions are essential in supporting the effective integration of weather data with Air Force TBM applications. The WxQuery functions take a pair of I-J coordinate values (entered by the application as latitude/longitude and converted by ConvertLatLonToIJCoords()) and look up the corresponding weather data value (or values in the case of multiple parameter data like winds) in the grid created by WxLoad(). The function then returns the weather data value at that point, or an interpolated value as defined by the WxContour routines in the case of a value between grid points. The WxTest prototype application demonstrates how WxQuery functions are used to automate the integration of weather data, by supplying inputs to a decision model incorporating data at a specified date and area of interest. Further development of WxQuery functions could support the rapid integration of arrays or tables of weather data as inputs to more sophisticated decision models. Note that WxQuery functions depend on the WxLib coordinate conversion functions to convert latitude/longitude to IJ values, and the coordinate conversion routines in turn rely on the application to supply the latitude/longitude point of interest.

6.6   Undeveloped Functions

The following functions in the WxTool rapid prototype were not documented or demonstrated in the WxLib and WxTest prototype, but could be included in future development:

6.6.1   Units Conversions - Routines to set default units of measure for weather data and also perform conversions (Kelvin to Fahrenheit, Feet/Second to Meters/Second, etc.).

6.6.2   Display Gridded Data Values at Grid Points - Shows the actual uniform grid numeric values. Similar to displaying contoured iso-lines, but shows individual point values.

Two important functions of a WxTK were either present in the WxTool but non-functional or undeveloped:

6.6.3   Display of Uniform Gridded Imagery Data: The WxTool has the procedure calls and interface to display Satellite Global Data Base (SGDB) Imagery data products, including visual and infrared images, but the functionality is not operational.

6.6.4   Drawing and Displaying Weather Symbology: An important component of the WxTK is a means for displaying the standard set of weather symbology on a map (Wind Barbs, Front Lines, Precipitation, etc.). A "Weather Graphics Editor" similar to the CMTK/MACS Graphics Editor is required to complete the functionality of the WxTK.

7.   Architecture of the WxAPI Function Library

The role of the WxLib functions in the architecture of the WxTK is depicted in Figure 4. By using the header files defined the Wx Include directory, the WxLib functions provide the link needed by GIS applications to integrate gridded weather data products with mapping and mission data. Notice the display function (in WxContour) relies on the particular mapping and display capability used by the application, and is therefore not a truly "common" toolkit function. All other functions should require only the use of the C programming language and the SUN/Unix system environment.

8.   Ambiguities and Conflicts Concerning the WxTK Functional Requirements

As experienced systems developers may expect, the number of ambiguities and conflicts that can be enumerated for a even a small scale project such as the WxTK can be enormous. Therefore, only the most crucial ambiguities and potential conflicts in the current state of WxTK development have been outlined here. The goal is to demonstrate how developers might begin to identify and track ambiguities that will lead to conflicts later. A comprehensive guide for

managing ambiguity and conflict is beyond the scope of this research effort, but hopefully a seed will be planted in the minds of developers and researchers for further interest and study.

8.1 The future of the Wx database design

The Air Force is currently developing a new centralized weather data distribution system for all TBM applications. The Combat Air Force Weather Support Package (CAFWSP) involves a relational database design for all AFGWC weather data using the ORACLE database management system software. Implementing a relational database design would change the current format of the uniform gridded database and require re-development of the WxDBRead
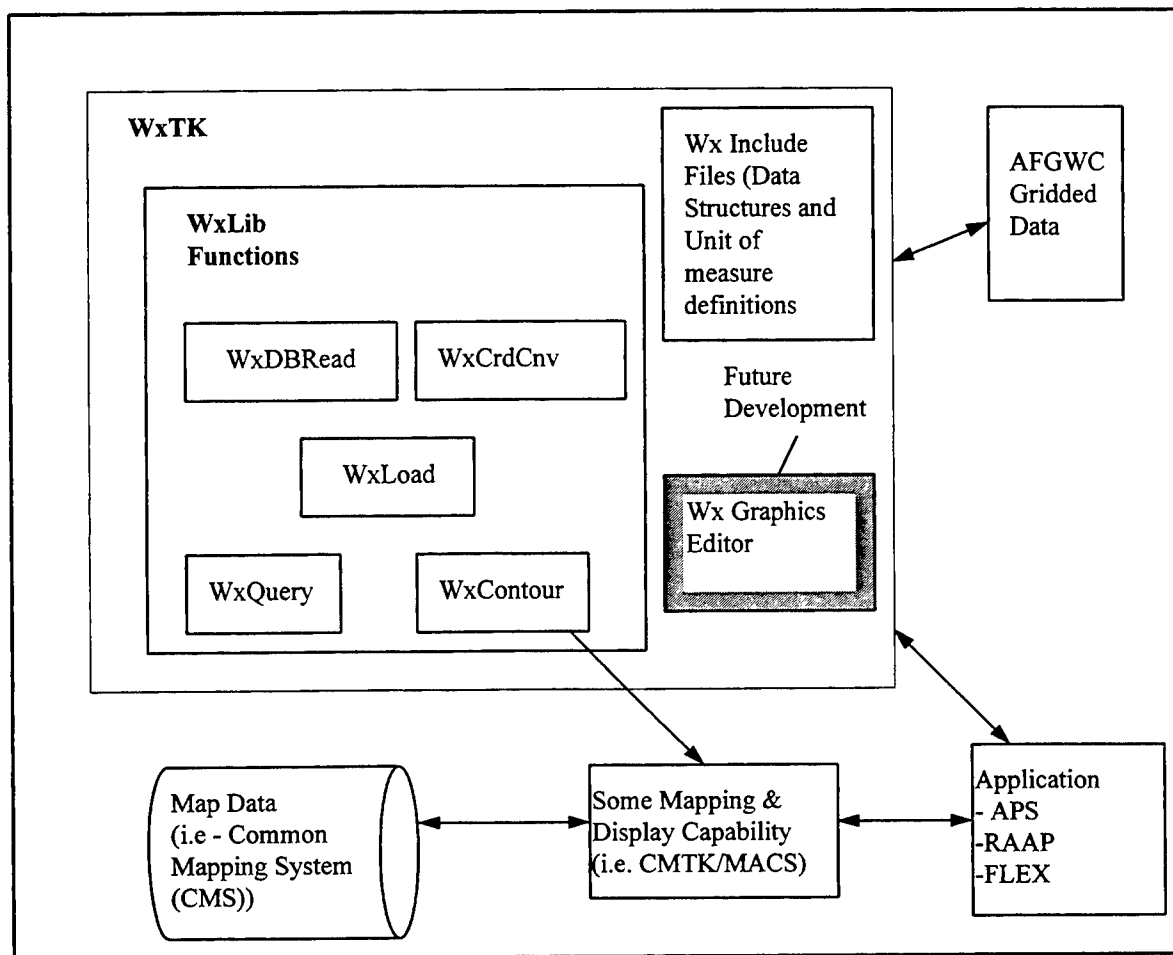
Figure 4
Proposed WxTK Architecture

and WxLoadData functions. Current progress and schedule for the project is unknown, as is the with the use of the WxTK. The sooner a precise definition of how data file formats will change can be made, the sooner WxLib functions can be re-designed and synchronized with the CAFWSP system.

8.2   Accuracy of the AFGWC Gridded Data

Another uncertainty involves the accuracy of AFGWC gridded weather data values at a particular point. Since gridded data are the results of model estimates and not actual readings, and interpolation is used for creating data between grid points, it remains unclear if user requirements are met given the current accuracy of AFGWC data. Insufficient accuracy of gridded data values will reduce incentive to use WxLib functions. Changes made to the models and interpolation routines may also affect WxLib contouring algorithms and methods of interpolation between grid points. Developers of the WxTK need to be aware of the potential conflict between users' desire for data accuracy versus AFGWC resources available for meeting those desires.

8.3   Application's Choice of Mapping Capability

Most of the current WxTK functions are defined in a generic sense, where use of a particular mapping capability by an application is not a factor. However, the WxLib functions as demonstrated in the WxTest Application have been developed and tested only with the CMTK/MACS. It is unclear whether the CMTK/MACS will always be the Air Force's "capability of choice". It is also uncertain whether the functions are truly generic and compatible with another mapping capability. Therefore, further development of the WxTK should include testing the generic WxLib functions with other mapping system software to ensure compatibility. Display functions have dependencies on the particular mapping system used, and functions to display data or pick a point from a map display will likely remain the responsibility of the application's mapping capability of choice.

8.4   System Hardware/Software Configurations

The current WxLib was developed on a SUN/SPARC station with the Sun/OS, X-Windows and Motif software systems. Future incompatibilities with the WxTK may result if the Air Force (or Air Force users) desire a different configuration for their applications (for example PC's and Windows/OS). The amount and cost of re-work required to alter the WxLib functions to fit

another hardware/software configuration is unknown. Testing the current generic WxLib functions on alternative system platforms may help developers anticipate potential conflicts.

8.5 Application Needs

Air Force GIS applications have different needs for weather data integration based on mission and planning horizons. For example, APS typically needs accurate weather data forecasts 36 to 48 hours ahead for advanced planning purposes, where FLEX needs the latest (or current) weather observations for last minute decisions. It is not clear whether the specific needs of TBM applications will conflict with each other or with the current WxLib functions. Involvement of application users in the development of the WxTK through evaluation of the WxTest Application and use of the WxLib functions (utilizing an evolutionary prototyping approach as described in {11}) is crucial to discovering ambiguities and potential conflicts.

9. Conclusions

Ambiguity and conflict in ISD become increasingly inevitable when developing "common" software tools like the WxTK, which serve applications with widely different missions and needs. Adding to the problem is the rapid advance of technology and volatility of mission environments. The challenge is to discover the common needs of all, and define functional requirements so each will not result in unanticipated conflict with the other, particularly if another part of the system changes (i.e. the AFGWC database design). Unanticipated conflict, resulting from functional requirements shrouded by ambiguity, may lead to higher costs of development in terms of resources spent on resolution. Meeting the challenge of unanticipated conflict requires consideration of potential diversity in weltanschauungen, in contrast to traditional monolithic approaches to ISD, and a significant amount of ambiguity and conflict management. The research describes how ambiguity emerges in functional requirements specification for the WxLib portion of the WxTK, and suggests issues to be defined and clarified in the early stages of development to avoid potentially costly conflicts in later phases of design and implementation.

References

{1} 7002-FTR-10 94-00. <u>Final Technical Report, Air Force Geographic Information Handling System</u> (AFGIHS), October 1994.

{2} Klein, H., H. Hirschheim and K. Lyytinen. <u>Conceptual and Philosophical Foundations of IS Development</u>, Unpublished textbook draft, 1993.

{3} Curtis, B., J. Elam and D. Walz. "Inside a Software Design Team: Knowledge Acquisition, Sharing, And Integration", <u>Communications of the ACM</u>, vol. 36, No. 10, October 1993.

{4} Gause, D. and G. Weinberg. <u>Exploring Requirements, Quality Before Design</u>. New York: Dorset House, 1989.

{5} Checkland, P., and J. Scholes. <u>Soft Systems Methodology in Action</u>. Chichester: John Wiley, 1990.

{6} Coser, L. <u>The Social Functions of Conflict</u>. Glencoe, Ill.: The Free Press, 1956.

{7} Brown, D. <u>Managing Conflict at Organizational Interfaces</u>. Reading, Mass: Addison-Wesley,1983.

{8} <u>Webster's NewWorld Dictionary, 2nd College Edition</u>. NY: Simon and Schuster, 1982.

{9} Boehm, B. <u>Software Engineering Economics</u>. Englewood Cliffs: Prentice-Hall, 1981.

{10} Hoaglin, D., and R. Light. <u>Data For Decisions</u>. Cambridge: Abt Books, 1982.

{11} Wetherbe, J. and N. Vitalari. <u>Systems Analysis and Design: Best Practices</u>. St. Paul: West Publishing, Fourth Ed., 1994.

{12} Connell, J. and L. Shafer. <u>Structured Rapid Prototyping An Evolutionary Approach to Software Development</u>. Englewood Cliffs, NJ: Prentice Hall, 1989.

{13} AFGWC. <u>AFGWC Dial-in Subsystem Software Users Manual, Version 3.2</u>. March 1995.

{14} USAF Environmental Technical Applications Center. <u>HIRAS USAFETAC Climatic Database Users Handbook No. 5</u>. October 1988.

Michael Callahan Report not available at time of publication.

Enhanced Graphical User Interface for Imagery Toolkit and Its Extensions

Shy-Shyan Chen

Ph.D. Candidate

Department of Computer and Information Science

New Jersey Institute of Technology

Newark, NJ 07102

Enhanced Graphical User Interface for Imagery Toolkit and Its Extensions

Shy-Shyan Chen

Ph.D. Candidate

Department of Computer and Information Science

New Jersey Institute of Technology

Abstract

For effective exploitation of digital imagery data, there is a great need for collecting image processing routines which can easily and effectively used on a variety of data. An Imagery Exploitation 2000 (IE2000) imagery toolkit, initially developed by professors Robert Stevenson and Robert Snapp, have been upgraded to SUN workstation runing Solaris. The graphical user interface, preliminarily developed by Audrey Copperwheat, have extensively been integrated for all the functions of imagery toolkit. It includes the ofter-used techniques in image processing such as image manipulation, enhancement, feture extraction, filtering, degradation, and statistics. We have also included some useful image processing rountines such as morphological operations and image segmentation.

Enhanced Graphical User Interface for Imagery Toolkit and Its Extensions

Shy-Shyan Chen

1. Introduction

The Imagery Exploitation 2000 (IE2000) imagery toolkit was initially developed at Rome Laboratory under the 1993 AFOSR summer faculty research program by professors Robert Stevenson [15] and Robert Snapp [13]. Then Audrey Copperwheat added graphical user interface for some functions. In this project, we have extensively implemented the graphical user interface for all the functions of imagery toolkit. Furthermore, some useful image processing routines such as morphological operations and segmentation are added.

The IE2000 Imagery Toolkit is a general purpose government-off-the-shelf imagery exploitation software package intended for distribution to Department of Defense (DOD) application developers and users. It was developed on SunOs 4 and we upgrade to SunOs 5 or Solaris. It was written in ANSI C to make portable and was developed using Motif. It is intended to be a comprehensive tool, covering all the fields in image processing. An effort to include most of techniques has been attempted, but due to limited time and its large number some remain to be implemented. Later versions will include missing techniques as well as new ones. The toolkit has aimed to make it very easy to upgrade and modify. Users can quickly learn about different techniques by inspecting the codes and can enhance the system by either implementing new techniques not provided or improving the ones provided.

2. The Main Window

The principal interface to the program is the main window. It mainly consists of three areas: the menubar, the editing icons and the display area. Under the display area, there are three scroll buttons to allow users changing zooming factor, image contrast, and viewing brightness. Fig. 1 shows the main window.

2.1. The menubar – There are 11 pulldown menus located in the menubar. The first one "File" is used to call read and write dialogs, and to clear, restore and exit. The following 8 menus carry out often-used image processing routines. The menu "Interface" intends to interact with other application programs. The "Help" menu provides on-line information about the application.

2.2. The editing icons – The editing icons give users easy access of actions including clockwise or counterclockwise rotation, image cutting, zooming, and audio interface.
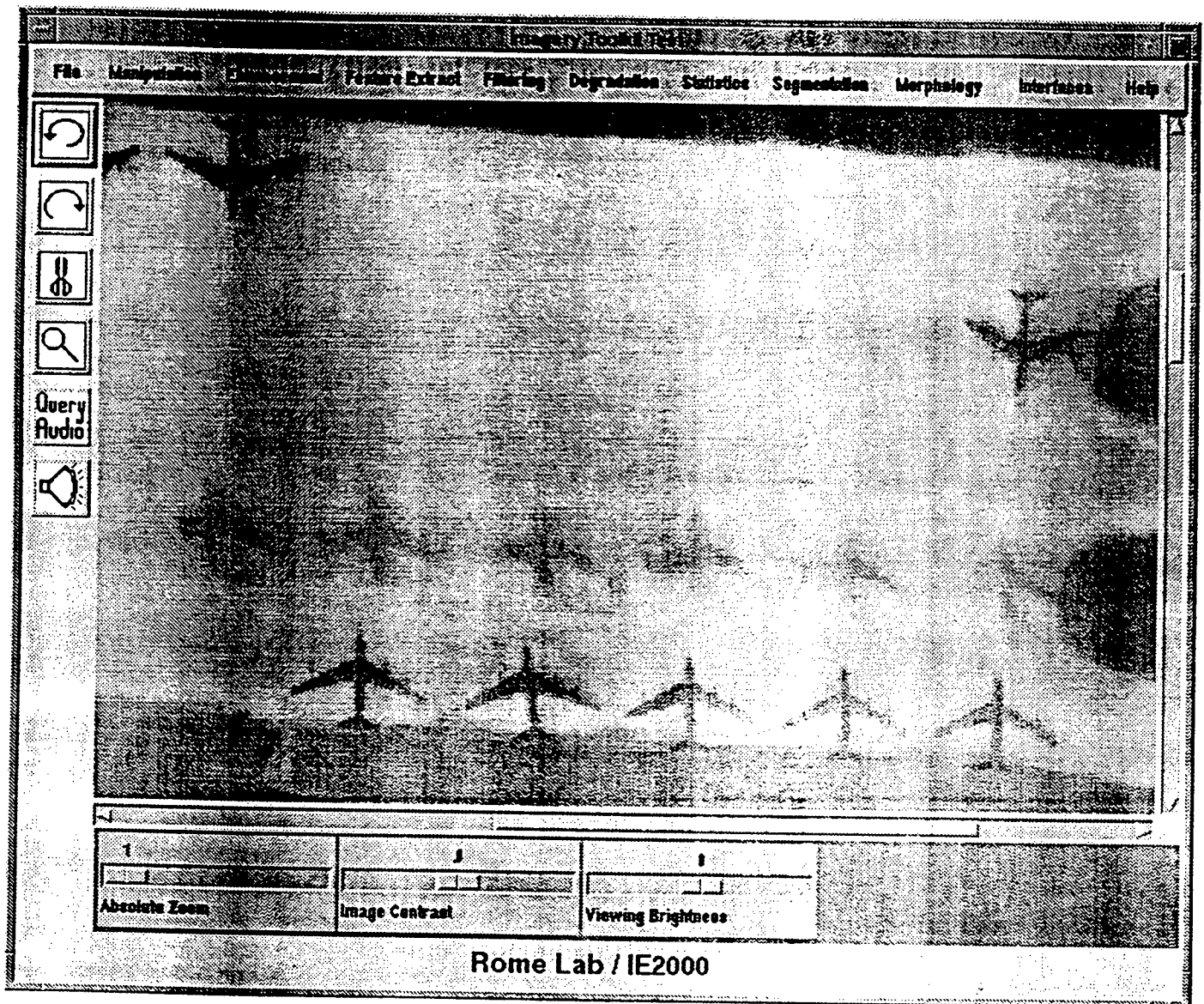
Fig. 1. The main window.

2.3. The display area – This area is reserved for image display. If the image size is larger than the display area, horizontal and vertical scroll bars are automatically shown.

3. The Read and Write Dialogs

These dialogs are used to select a file to read or write an image from or to an imagery server or disk. Both use the file selection box widget which encapsulates the task of opening a directory file, reading its entries and traversing the directory tree. The contents are displayed in two list widgets: one holds the directory files within a given directory and the other holds ordinary, link and device files. Selecting a file and traversing the directory tree are done by clicking an item in the appropriate list widget. The format supported currently is TIFF. Fig. 2 shows the read/write dialog.
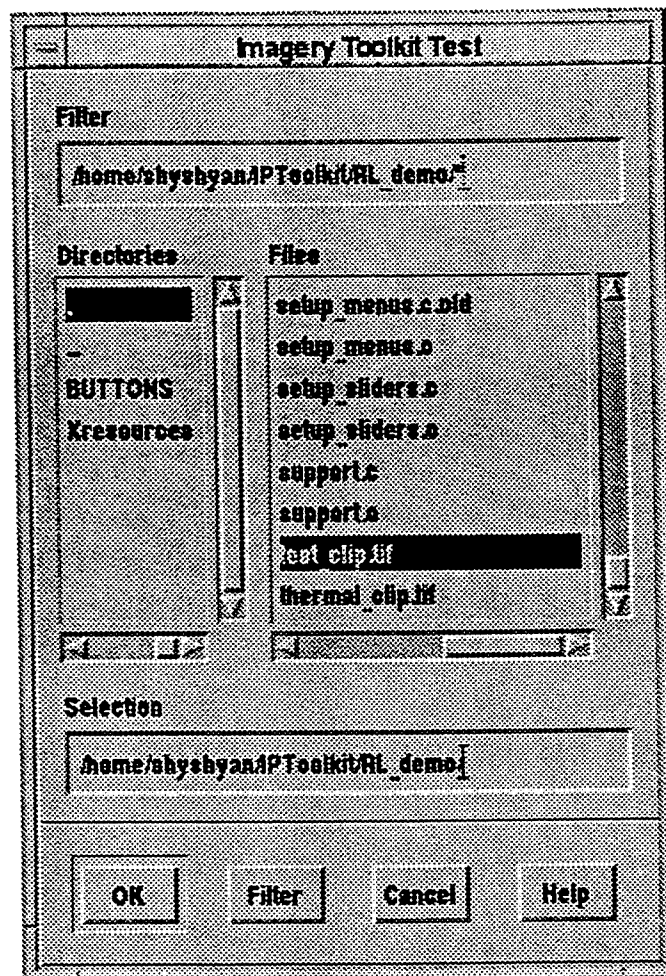
Fig. 2. The read/write dialog.

4. The Image Processing Routines

The image processing routines are grouped into 8 categories: manipulation, enhancement, feature extraction, filtering, degradation, statistics, segmentation, and morphology.

4.1. Manipulation – The image manipulation contains zoom, rotation, crop, flip on the horizontal or vertical axis, show an overview window, set parameters, and add or multiply a constant. Fig. 3 shows the munipulation pulldown menus. The "Zoom" menu includes 3 pulldown menus: zoom in, zoom out, and original image restoration. The zoom factor can be set in the menu of "Set Parameters." The "Rotate" menu includes 3 pulldown menus: rotate right (clockwise), rotate left (counter-clockwise), and restore to 0 rotation (the original image). The rotation angle (degree) can also be set in the menu of "Set Parameters." The "Show Overview Window" allows the creation of another window showing the image processed. This makes it possible to compare serveral images processed by different techniques. The "Add Constant" menu calls a pulldown

menu: constant to add. It is possible to increment each pixel with a given amount to alter the brightness. Similarly, each pixel can be multiplied by a given ratio to alter the contrast by using the "Multiply Constant" menu.

Fig. 3. The manipulation pulldown menus.

4.2. Enhancement – The image enhancement contains adjust mean, adjust contrast, adjust Gamma, map intensities, invert intensities, stretch range, mask image, set parameters, and histogram equalization. Fig. 4 shows the enhancement pulldown menus. The contrast, Gamma, and intensity offset can be selected by using the "Set Parameters" menu. The "Mask Image" is to enhance an image using an unsharp mask operator whose window width and height and the alpha value used as the fraction of the highpassed added to the original, can also be selected by using the "Set Parameters" menu.

4.3. Feature Extraction – The feature extraction contains threshold count, etc., edge detection, and set parameters. Fig. 5 shows the feature extraction pulldown menus. The "Threshold Count" returns the number of pixels above threshold or below threshold. The "etc." reserves for future extension. The "Edge Detection" includes the Canny, Frei and Chen, Kirsch, Marr Hildreth, Prewitt, Roberts X, Robinson, and Sobel edge operators.

4.4. Filtering – The image filtering contains maximum, average, minimum, cross median, square median, cross mean trimmed, square mean trimmed, convolution, and Gaussian smoothing. Fig. 6 shows the filtering pulldown menus. The "Convolution" menu displays a pulldown menu which allows the interactive input of a 3 × 3 mask. The "Gaussian Smoothing" menu allows the input of standard deviation. The remaining filters allows the selection of either 3 × 3 or 5 × 5 window.
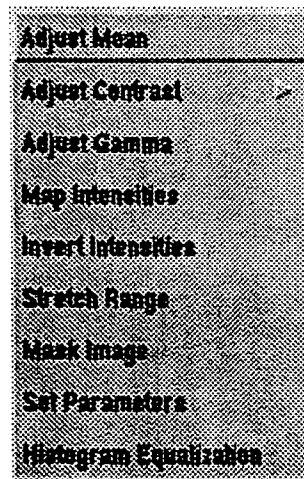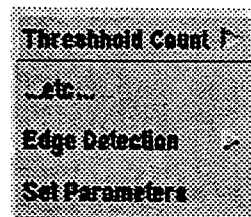
Fig. 4. The enhancement pulldown menus.

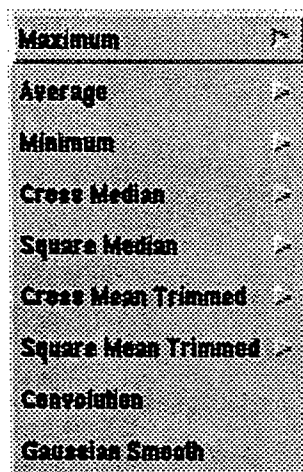Fig. 5. The feature extraction pulldown menus.

Fig. 6. The filtering pulldown menus.

4.5. Degradation – The image degradation contains random noise and uniform noise additions. Fig. 7 shows the degradation pulldown menus. The random noise corrupts individual bits in the image with the probability "percentage." The "seed" parameter initializes the random number

generator. The uniform noise is distributed uniformly over the range from -range to +range. The "seed" parameter initializes the random number generator.
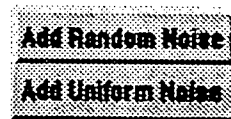
Fig. 7. The degradation pulldown menus.

4.6. Statistics – The image statistics contains moment, standard deviation, maximum, mean and minimum. Fig. 8 shows the statistics pulldown menus. These buttons return the statistical values measured for the image.
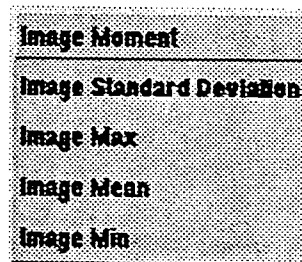
Fig. 8. The statistics pulldown menus.

4.7. Segmentation – The image segmentation contains bilevel threshold, half threshold, and multilevel threshold. Fig. 9 shows the segmentation pulldown menus. The "Bilevel Threshold" creates a binary image by mapping all pixel values below "Threshold" to 0 and all above to MAXPIXEL. The "Half Threshold" creates a grayscale image by mapping all pixel values above "Threshold" to MAXPIXEL and all below to their original values. The "Multilevel Threshold" creates a four-level image by selecting three threshold values and mapping all pixels values in between to 0, 85, 170, and 255, respectively.
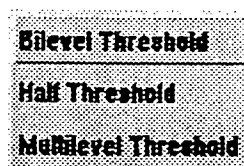
Fig. 9. The segmentation pulldown menus.

4.8. Morphology – The mathematical morphology contains binary dilation, erosion, opening, closing, soft dilation, soft erosion, soft opening, soft closing, set parameters, grayscale dilation,

and grayscale erosion. Fig. 10 shows the morphology pulldown menus. The binary morphological opertors allow the selection of a 3×3 or 5×5 flat structuring element. The soft morphological operators besides the same selection have the input "rank" set up in the "Set Parameters" menu. The grayscale morphological operators allow the input values of a 3×3 structuring element.
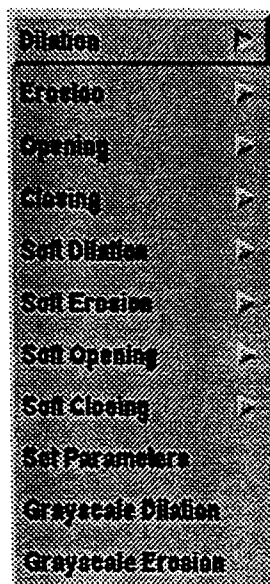

Fig. 10. The morphology pulldown menus.

References

[1] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 532-550, July 1987.

[2] F. Y. Shih and O. R. Mitchell, "Threshold decomposition of grayscale morphology into binary morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 31-42, Jan. 1989.

[3] R. R. Snapp, *IPToolkit: An Image Processing Environment for the X Window System*, Final Report for Summer Faculty Research Program, Rome Laboratory, Aug. 1993.

[4] R. L. Stevenson, *Image Processing Toolkit*, Final Report for Summer Faculty Research Program, Rome Laboratory, Aug. 1993.

Appendix A
Image Segmentation Routines

NAME

IPHalfThre - threshold an image to create a half-thresholded image

SYNOPSIS

#include <IPTip.h>

IMAGE *IPHalfThre(in, Threshold)
IMAGE *in;
PIXEL Threshold;

DESCRIPTION

IPHalfThre() creates an image by mapping all pixel values above *Threshold* to MAXPIXEL and all below *Threshold* keep unchanged. The processed image is returned.

NAME

      IPMultThre - threshold an image to create a 4-level image

SYNOPSIS

      #include <IPTip.h>

      IMAGE *IPMultThre(in, Threshold1, Threshold2, Threshold3)
      IMAGE *in;
      PIXEL Threshold1, Threshold2, Threshold3;

DESCRIPTION

      IPMultThre() creates a 4-level image by mapping all pixel values below *Threshold1* to 0, pixel values between *Threshold1* and *Threshold2* to MAXPIXEL / 3, pixel values between *Threshold2* and *Threshold3* to 2 * MAXPIXEL / 3, and all above to MAXPIXEL. The processed image is returned.

SYNCHRONIZATION OF HARMONICALLY MODELOCKED ERBIUM FIBER LASERS

Walter Kaechele
Graduate Student
Physics Department

Rensselaer Polytechnic Institute
Troy, NY 12180

SYNCHRONIZATION OF HARMONICALLY MODELOCKED ERBIUM FIBER LASERS

Walter Kaechele
Graduate Student
Physics Department
Rensselaer Polytechnic Institute

Abstract

The synchronization of two erbium doped fiber lasers of differing configurations was experimentally studied. An actively modelocked ring lasers served as the synchronizing source at various frequencies. This signal was introduced to a linear fiber cavity with a multi-quantum well saturable absorber acting as the modulating device. Evidence of synchronization is presented as well as the presence of Q-switching in the passively modelocked system.

SYNCHRONIZATION OF HARMONICALLY MODELOCKED ERBIUM FIBER LASERS

Walter Kaechele

Introduction:

High speed optical fiber communication systems utilizing all-optical processing should allow data signal generation into the terabit per second regime. Manipulation of data at such high speeds requires development of novel transmission and reception technologies. A prominent reception issue is clock recovery, meaning a continuous stream of pulses synchronized with the incoming data stream can be resolved from that same data stream. We investigate experimentally the synchronization of two erbium fiber lasers.

Fiber systems have been synchronized utilizing both the Kerr nonlinearity or cross-phase modulation [1-3] and multi-mode modulation [4]. We modulated the passive system using the former, but rather than utilizing a passive ring configuration, we examined synchronization in a linear cavity employing a multi-quantum well substrate as a saturable absorber. We report results at the fundamental cavity period and the tenth multiple of that period.

Experimental Details:

Figure 1 shows the experimental setup for the system. The input signal was generated using an actively modelocked fiber ring laser tuned so that the cavity spacing matched that of the MQW laser system. The wavelength of the input signal was tuned so that its spectrum did not overlap any part of the MQW laser's modelocked spectrum. This also allowed us to filter out the input signal from the MQW lasers output.
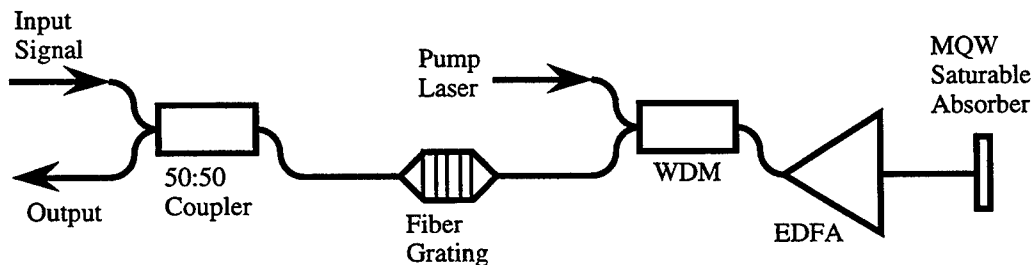
Figure 1: Experimental setup of the MQW synchronized laser. The input signal comes from the ring laser, and the combined output passes out the same coupler. WDM = wavelength division multiplexer, EDFA = erbium doped fiber amplifier.

The MQW laser was set up with a fiber Bragg grating serving as one reflector and a multi-quantum well saturable absorber as the other. The fiber grating reflected at better than 80% in an eight nm window centered at 1.558 µm, and was transparent to pulses at other wavelengths, such as the injection signal. The MQW saturable absorber was butt coupled against a cleaved fiber end so as to complete the cavity and allow stable modelocking. The MQW saturable absorber was composed of 50 alternating layers of AlInAs and GaInAs each having a thickness of 20 nm deposited on an InP substrate. The transmission spectrum of the well mirror is shown in figure 2. With the erbium amplifier pumped by a 980 nm diode laser. this system modelocked at 3.398 Mhz.
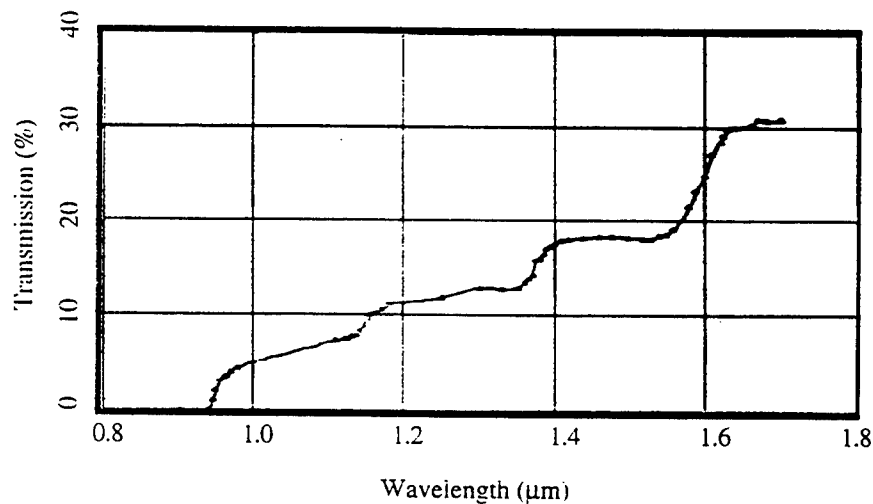

Figure 2: *The transmission spectrum of the MQW saturable absorber. The lasers operated in the 1.5 - 1.6 µm region. where the transmission is nearly constant at 17%.*

The input signal was introduced to the MQW laser through the fiber grating and allowed to make one pass through the laser cavity before leaving at the fiber grating. This was the signal to which the MQW laser synchronized. To observe synchronization the combined signal from the MQW laser output was sent through a filter to remove any residual input pulses. The MQW laser's signal was then sent to a photodetector along with a separate signal from the source laser. so that both could be viewed in synchronization with one another on an oscilloscope as shown in figure 3.

Figure 3: *Oscilloscope trace of the synchronized output at the fundamental frequency of 3.3898 MHz. The smaller pulses are the input signal from the actively modelocked ring laser, while the larger pulses are the MQW laser's synchronized output.*

Prior to synchronization, both lasers put out similar pulse trains with a large cw background as seen in figure 4. With the MQW laser set below the modelocking threshold, a very small signal would induce the MQW laser to modelock although it would not be synchronized. With the MQW laser already in a stable modelocking state, a high input signal power (>2mW) was necessary so as to intrduce a large enough modulation in the fiber's nonlinear index. When the input signal was removed from the MQW laser, the MQW laser failed to return to its previously modelocked state.
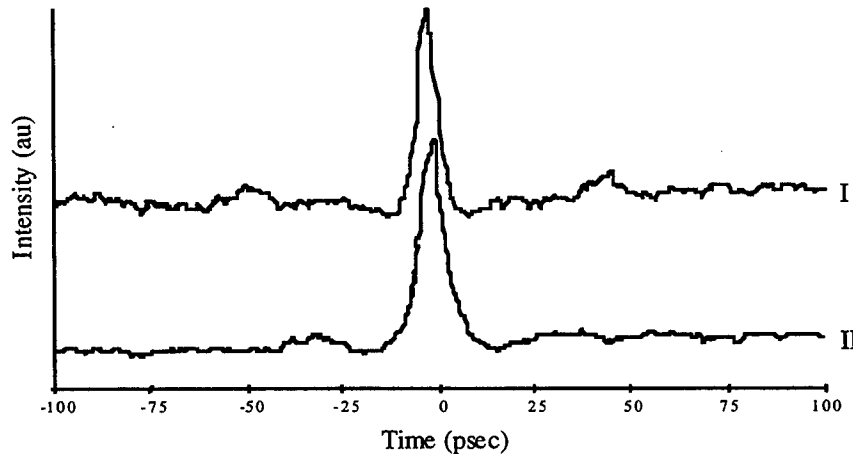
Figure 4: *Autocorrelation traces of the modelocked pulses from the ring laser (I), and the MQW laser (II).*

4-5

Results:

With the signal pulses injected into the MQW laser cavity, the lasers were synchronized to the frequency of the input pulses so long as the bit rate was a multiple of the MQW laser's fundamental cavity frequency. The output pulses from the synchronized system seen in the inset of figure 5 show the output pulses at the tenth multiple of the MQW laser. Every tenth pulse is considerably larger than the others, indicating that the laser still favors the fundamental cavity frequency. In addition a rise in amplitude can be seen in the larger pulses in the inset of figure 5. This rise in the pulses corresponds to a Q-switching effect occurring in the synchronized output. The Q-switched envelope is approximately 40 nsec in width with a repetition rate of 8 kHz. This modulation is possibly due to a relaxation interaction occurring between the MQW and the gain medium, as no evidence of Q-switching was seen in either laser before injection.
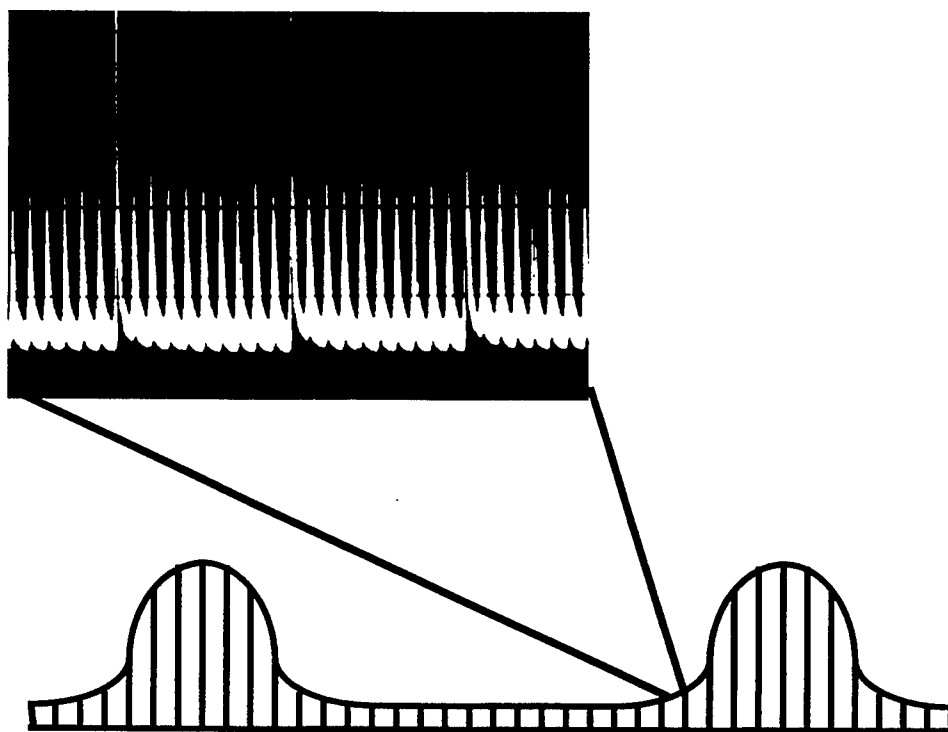

Figure 5: *Modelocked/Q-switched output of synchronized laser. The small solid lines are the modelocked pulses while the large envelope on top represents the Q-switching. The inset shows an oscilloscope trace of the synchronized laser at the tenth harmonic. The larger pulses show the beginnings of a Q-switched pulse.*

This additional modulation thwarted efforts to obtain an accurate measurement of the synchronized pulse width. The spectrum of the synchronized laser shown in figure 6 exhibits spectral narrowing from the original modelocked spectrum, with some additional
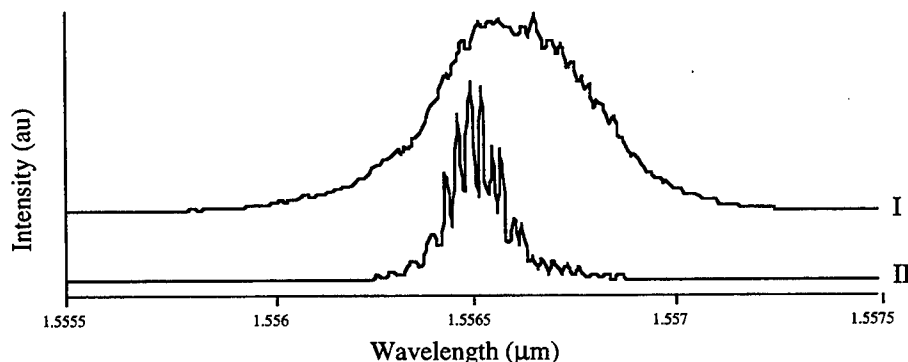
Figure 6: *Spectra from the MQW laser. (I) shows the original modelocked spectra from the laser prior to the signal being injected. (II) displays the pulse narrowing after synchronization as well as the additional modulation of the spectra.*

modulation due most likely to the Q-switching. The RF spectrum also showed this modulation effect. Figure 7 shows the RF spectrum of the MQW laser prior to synchronization and the synchronized output. The additional sidebands on the synchronized signal are separated by approximately 8 kHz, the rate at which Q-switching occurs, indicating that it is indeed a product of the modelocking/Q-switching interaction.
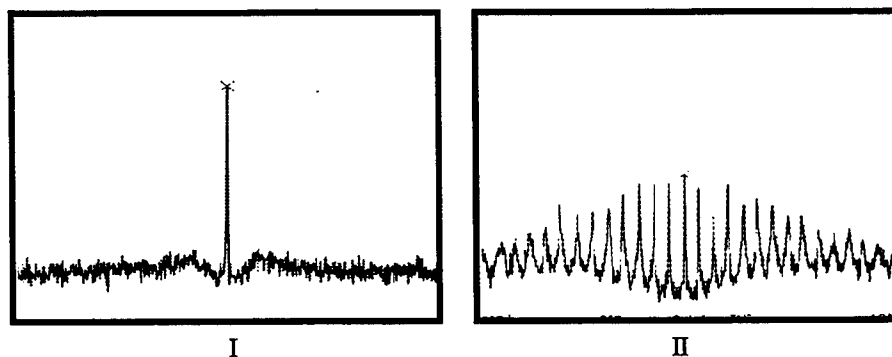
Figure 7: *RF spectra of the tenth harmonic sideband of the MQW laser in a modelocke state (I), and the same laser in the synchronized state (II). The most notable differences are the decrease in amplitude of the central peak and the additional spikes in the synchronized spectrum.*

Conclusion:

Synchronization between fiber lasers of differing geometries has been demonstrated. Additional interactions between the gain medium, saturable absorber and input signal has also been observed. The output pulses from the synchronized laser exhibited a modelocked/Q-switched behavior at both driving frequencies. Additional investigation of the dynamics causing this effect is being carried out to examine how higher repetition rates (exceeding the relaxation rate of the MQW saturable absorber) will behave.

Acknowledgments:

References:

1   Smith, K., and Lucek, J. K.: 'All-optical clock recovery using a modelocked laser', *Electron. Lett.*, 1992, **28**, pp. 1814-1815

2   Øbro, M., Thorsen, P., and Andreasen, S. B.: 'All-optical frame synchronisation recovery', *Electron. Lett.*, 1994, **30**, pp. 1243-1244

3   Ellis, A. D., Smith, K., and Patrick, D. M.: 'All optical clock recovery at bit rates up to 40 Gbit/s', *Electron. Lett.*, 1993, **29**, pp. 1323-1324

4   Margalit, M., Orenstein, M., and Eisenstein, G.: 'Noise reduction by harmonic injection locking of passively modelocked erbium-doped fiber lasers', *Optics Lett.*, 1995, **20**, pp. 1877-1879

4-8

GENERATION AND APPROXIMATION OF SPHERICALLY INVARIANT RANDOM VECTORS

Andrew D. Keckler
Doctoral Candidate
Department of Electrical Engineering

Syracuse University
Department of Electrical and Computer Engineering
121 Link Hall
Syracuse, New York 13244

Generation and Approximation of Spherically Invariant Random Vectors

Andrew D. Keckler
Syracuse University, Department of Electrical and Computer Engineering
121 Link Hall, Syracuse, NY 13244

ABSTRACT

The spherically invariant random vector (SIRV) model is used to characterize correlated, non-Gaussian clutter samples. In order to evaluate simulations using this model, an efficient technique for the generation of SIRV's is required. Through the use of the representation theorem for SIRV's, a SIRV can be generated from the product of an univariate random variable and an independent Gaussian distributed vector. An automated approach is presented for the generation of univariate data via the generalized rejection theorem, and the results are extended to the generation of SIRV's. A technique for approximating a SIRV using a multivariate Gaussian-mixture distribution is proposed, and a parameterized receiver based upon this approximation is presented.

Keywords: spherically invariant, random vectors, non-Gaussian clutter, random number generation, rejection theorem, Gaussian mixture

Generation and Approximation of Spherically Invariant Random Vectors

Andrew D. Keckler

1. INTRODUCTION

Conventional detection schemes are based upon the assumption of Gaussian distributed interference. In practice, however, a significant portion of clutter samples from a surveillance volume are often non-Gaussian, as has been describe by D.L. Stadelman and D.D. Weiner[10] and M. Rangaswamy, D.D. Weiner, and A. Ozturk[3]. This is particularly true for data collected at low grazing angles or at high resolution. Distributions such as the Weibull or K-distribution, which fall within a general class of distributions known as Spherically Invariant Random Vectors (SIRV's), are typically used to model real non-Gaussian radar clutter data. These PDF's typically have higher tails than the Gaussian PDF, which leads to more frequent occurrences of large clutter returns. These distributions are often described as "spiky", and this effect is illustrated in figure 1.1[10].
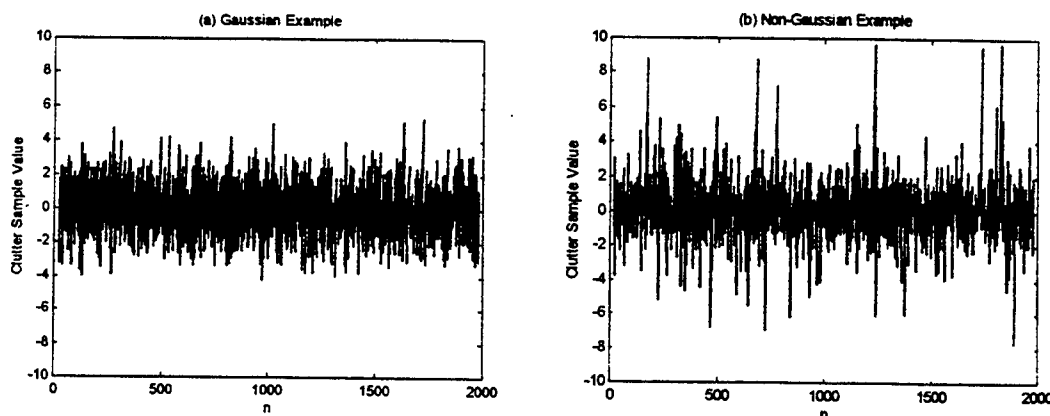

Figure 1.1: Time Sequence of "Spiky" Clutter Data.

Gaussian receivers tend to perform poorly in this environment. The extended tails of the non-Gaussian distributions require that the threshold of the Gaussian receiver be raised in order to maintain the desired false-alarm rate for the non-Gaussian clutter, which consequently causes a reduction in the probability of detection. In contrast, the optimal non-Gaussian receiver is found to contain a nonlinearity which reduces large clutter spikes, allowing the threshold to be maintained at lower levels[10], which in turn provides increased target detection opportunities at the desired false alarm rate.

The optimal non-Gaussian receivers require the specification of a suitable non-Gaussian PDF model. Generally these multivariate non-Gaussian models have been difficult to work with when the data is correlated. It has been found to be difficult to independently control the shape of the distribution and its covariance matrix. Spherically invariant random vectors (SIRV's) are a class of non-Gaussian distributions that allow for the independent control of the form of the distribution and its covariance matrix, making them suitable as a model for the interference. Closed form expressions exist for the multivariate PDF of many types of SIRV's, and the class includes many distributions of interest, such as the Student-t, the Weibull, and the K-distributions. Furthermore, the Gaussian distribution is a member of this class.

According to the representation theorem and the closure property of SIRV's to linear transformations, SIRV's can be generated as the product of an univariate random variable and a Gaussian random vector with zero-mean, independent components. The generation of independent zero-mean Gaussian distributed random variables is well understood. The univariate multiplier is used to control the shape of the SIRV distribution, and its PDF, known as the characteristic PDF, may well have a mathematically complex form that does not lead to a closed form for its cumulative distribution function (CDF), or the inverse of the CDF. The inverse of the CDF is required to readily generate data from this distribution. Many approaches exist to find approximations for the inverse CDF,

but the errors involved, especially in the tails of the distribution, may loom large in the evaluation of the non-linear, non-Gaussian receivers. If a function can be found that bounds the characteristic PDF from above, and for which an inverse CDF can be found, then the generalized rejection theorem can be used to generate data from the desired distribution without approximation error. The bound must closely fit the characteristic PDF, or a large percentage of the generated data points will be rejected, wasting time and depleting the available pool of random numbers.

In many cases, it may be difficult to find a suitable bound that closely fits the characteristic PDF. It is possible, however, to segment the PDF into a number of regions and to bound each region by a horizontal line. This leads to a continuous, piecewise linear CDF which is easily invertible. By increasing the number of segments, this bound can be brought arbitrarily close to the desired PDF. If the domain of the characteristic PDF is infinite, it becomes impractical to fit this bound across the entire domain. Instead, the piecewise linear bound is fit only to the "body" of the distribution, and the Cauchy PDF is used to bound the infinite tails. The Cauchy distribution has infinite variance, and its tails approach zero at a rate proportional to $1/x^2$. Thus it will bound the tails of most other PDF's, although loosely. Furthermore, Cauchy distributed random variables are easily generated.

The PDF of all SIRV's can be expressed in terms of a quadratic form q(y) given by

$$q(y) = (y\text{-}b)^H \Sigma^{-1}(y\text{-}b) \qquad (1.1)$$

where Σ is the covariance matrix and **b** is the mean vector of the SIRV **y**. Often the problem of interest can be cast in terms of the quadratic form only. Since the quadratic form is always a scalar, it can be generated directly as a univariate random variable, and the above approach can be applied directly. This avoids the numerous problems associated with generating uncorrelated Gaussian random vectors.

The characteristic PDF of SIRV's commonly considered, such as the Weibull or K-distributions, are continuous functions. This is not a requirement of the SIRV model, however. The discrete Gaussian-mixture SIRV[4,7] has the characteristic PDF,

$$f_s(S) = \sum_{k=1}^{K} w_k \delta(S\text{-}s_k), \quad for \ w_k > 0 \quad and \ \sum_{k=1}^{K} w_k = 1. \qquad (1.2)$$

Through the proper choice of the parameters in equation (1.2), the discrete Gaussian-mixture SIRV can approximate many other types of SIRV's. This has many advantages, because the distribution of the discrete Gaussian-mixture SIRV is simple to evaluate, as is the distribution of its quadratic form. Random samples can easily be generated for the discrete Gaussian-mixture SIRV, and the form of the PDF for the discrete Gaussian-mixture SIRV leads to a parameterized receiver structure. This receiver can be used in place of the optimal receiver for any SIRV that the discrete Gaussian-mixture SIRV can approximate, without changing the structure of the receiver. Only the parameters need be adjusted.

2. SUMMARY OF THE SIRV MODEL

A brief review of the pertinent properties of the SIRV model is presented. The work of Yao [1] gives rise to a representation theorem for SIRV's that can be stated as follows:

Theorem 1: If a random vector is a SIRV, then there exists a non-negative random variable s such that the PDF of the random vector conditioned on s is a multivariate Gaussian PDF.

A spherically invariant random vector **x** with N zero-mean uncorrelated elements can therefore be represented by

$$x = sz, \qquad (2.1)$$

where z is a Gaussian random vector with N zero-mean independent components, and s is an independent, non-negative random variable with PDF $f_s(S)$. This PDF uniquely determines the SIRV, and is called the characteristic PDF of the SIRV. It can be normalized such that $E(s^2) = 1$ without loss of generality [2].

A second property of SIRV's allows correlation to be introduced by a linear transformation, as stated in Theorem 2 [2]:

Theorem 2: If x is an SIRV with characteristic PDF $f_s(S)$, then

$$y = [A]x + b \qquad (2.2)$$

is also an SIRV with the same characteristic PDF, as long as the matrix [A] is nonsingular and b is a known vector having the same dimension as x.

The transformed vector y will have a mean vector b and a covariance matrix $\Sigma = [A][A]^H$. The PDF of the SIRV is the joint PDF of its N components and is given by

$$f_y(Y) = (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} h_N(Q(Y)) \qquad (2.3)$$

where Q(Y) is the quadratic form

$$Q(Y) = (Y-b)^H \Sigma^{-1} (Y-b) \qquad (2.4)$$

and $h_N(Q(Y))$ is a positive, real-valued, monotonic decreasing function given by

$$h_N(Q(Y)) = \int_0^\infty S^{-N} e^{-\left(\frac{Q(Y)}{2S^2}\right)} f_s(S)\, dS \qquad (2.5)$$

It can be seen that any N-dimensional SIRV is uniquely determined by its covariance matrix, mean vector and either its characteristic PDF or $h_N(Q(Y))$. Furthermore, the PDF of the quadratic form Q(Y) is given by

$$f_q(Q(Y)) = \frac{1}{2^{\frac{N}{2}} \Gamma(\frac{N}{2})} Q(Y)^{\frac{N}{2}-1} h_N(Q(Y)), \quad 0 \le Q(Y) \qquad (2.6)$$

This has an important implication. Since $h_N(Q(Y))$ uniquely identifies each type of SIRV, this indicates that the multivariate problem can be reduced to an equivalent univariate problem in many instances. Since the quadratic form q(y) is an univariate random variable, it can be generated directly with the techniques proposed here.

3. ACCEPTANCE-REJECTION METHODS

A brief review of the rejection theorem is presented. The rejection theorem can be stated as:

Theorem 3: Let s be a random variable with density $f_s(S)$ and r be any random variable with density $f_r(R)$ such that $f_s(S)=0$ whenever $f_r(R)=0$. Then let u be uniformly distributed on the interval (0,1). If r and u are statistically independent and

$$\eta = \{U \le T(R)\} \qquad (3.1)$$

where

$$T(R) = \alpha f_s(R)/f_r(R) \le 1, \qquad (3.2)$$

then the rejection theorem states

$$f_{r|\eta}(R|\eta) = f_s(R). \qquad (3.3)$$

The density $f_r(R)$ approximates $f_s(S)$ if the value

$$\alpha = \max_s f_s(S)/f_r(S) \qquad (3.4)$$

is a constant close to 1. If α equals 1, then f_s is identical to f_r. In figure 3.1, the function αf_r is seen to bound f_s in the sense that $\alpha f_r(S) \geq f_s(S)$ for all S in the support of s. It is desired to generate a random variable s
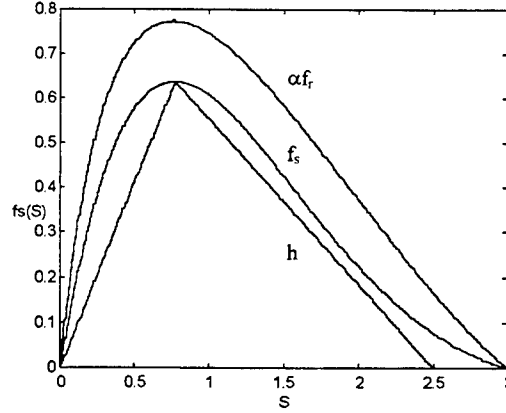

Figure 3.1: Illustration of the Acceptance-Rejection Method.

with density $f_s(S)$ from variates generated from $f_r(R)$. This can be accomplished using following algorithm[2]:

> STEP 1. Generate R from $f_r(R)$ and compute $T(R) = f_s(R)/\alpha f_r(R)$,
> STEP 2. Generate U as a random variable distributed uniformly
>   over the interval (0,1).
> STEP 3. If $U > T(R)$, reject R and return to STEP 1, else accept
>   R as a variate from $f_s(S)$.

If $f_s(S)$ is a time-consuming function to evaluate, and there exists a function h such that $h(S) \leq f_s(S)$ for all S in the support of s, then a fast, preliminary test can be made, as can be seen if figure 3.1. The modified procedure becomes:

> STEP 1. Generate R from $f_r(R)$ and compute $T_h(R) = h(R)/\alpha f_r(R)$,
> STEP 2. Generate U as uniform (0,1),
> STEP 3. If $U \leq T_h(R)$, accept R as a variate from $f_s(S)$,
> STEP 4. Else, compute $T(R) = f_s(R)/\alpha f_r(R)$. If $U \leq T(R)$, accept R
>   as a variate from $f_s(S)$, else reject R and return to STEP 1.

This procedure has a geometric interpretation. A point (R,Y) is generated in the region bounded by $\alpha f_r(R)$ and the R-axis. If the point falls within the region bounded by h(R) and the R-axis, accept R immediately. If not, then if the point falls within the region bounded by $f_s(R)$ and the R-axis, accept R. Otherwise reject R. The parameter α equals the area under the bound function, and the average efficiency of the acceptance-rejection algorithm is equal to $1/\alpha$.

4. GENERATION OF BOUNDS

In general, it is possible to partition a probability density function into M equal intervals such that horizontal line segments can be used to approximate the PDF, as shown in figure 4.1. It is assumed that the PDF is bounded and that the number of intervals is sufficient such that only one minima or maxima occurs in each interval. An optimization routine is used to find the maxima in each interval, which is designated as fu_k. The equation for the bound function in this region is

$$\alpha f_r(S) = fu_k, \quad S_0 + (k-1)\Delta S \leq S < S_0 + k\Delta S, \quad 1 \leq k \leq M, \qquad (4.1)$$

where ΔS is the width of each interval, which is found to be

$$\Delta S = (S_1 - S_0)/M. \qquad (4.2)$$
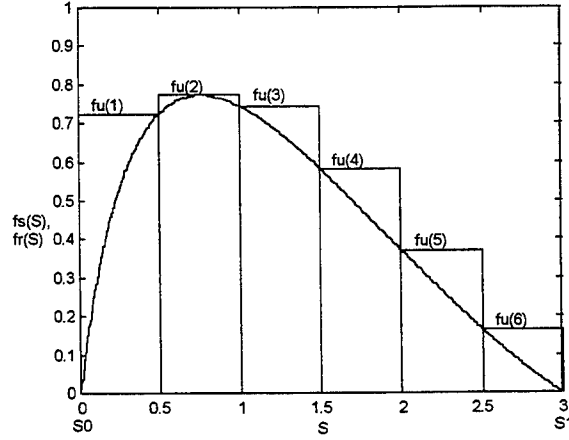
Figure 4.1: Piecewise Linear Bound of PDF.

Obviously, if the density has an infinite tail, the entire support cannot be segmented. The PDF is divided into a "body" and tail portion at a point sufficiently far into the tail. This is illustrated in figure 4.3 at the point S_1. Random variates can easily be generated for the Cauchy distribution, which is given by

$$f_x(X) = \frac{b}{\pi(b^2 + (X - \mu)^2)} \qquad (4.3)$$

and is plotted in figure 4.2. The Cauchy distribution is used to bound the tail portions of the PDF. It is assumed that the tail of the PDF $f_s(S)$ smoothly approaches zero at a rate faster than $1/S^2$. The location of the global maximum, S_{max}, of the body region of the PDF is used as the mean of the Cauchy distribution, and the shape parameter is chosen so that the Cauchy distribution intersects the piecewise linear bound at

Figure 4.2: PDF of the Cauchy Distribution.

Figure 4.3: Cauchy Bound of PDF Tail.

the endpoints of the "body" region. For a left tail, which is valid for $R < S_0$, the shape parameter is given by

$$b_1 = \frac{1 - \sqrt{1 - 4\pi^2 \, fu_1^2 (S_0 - S_{max})^2}}{2\pi \, fu_1} \qquad (4.4)$$

and for the right tail, which is valid for $R \geq S_1$, the shape parameter is

$$b_r = \frac{1 - \sqrt{1 - 4\pi^2 \, fu_M^2 (S_1 - S_{max})^2}}{2\pi \, fu_M} \qquad (4.5)$$

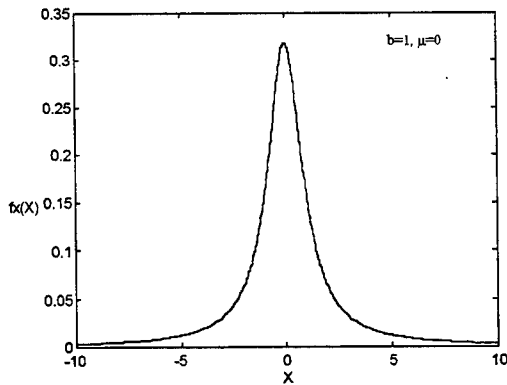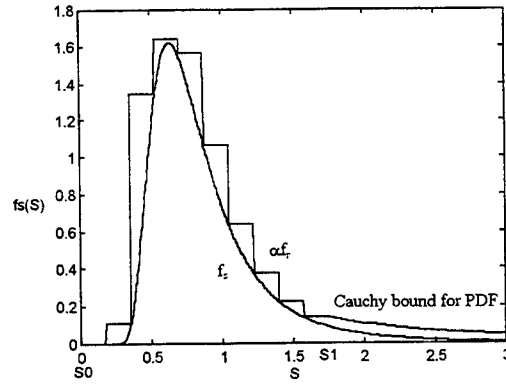Occasionally, the endpoints of the body region will be chosen too close to the location of S_{max} and the term within the square root will become negative. In this instance, either the declared body region of the PDF $f_s(S)$ can be expanded, or the square root term can be set to zero. If the square root is set to zero, there will be a jump in the bound function at the edges of the body region, leading to inefficiencies in the implementation of the acceptance-rejection algorithm. Furthermore, the means of the Cauchy distributions will shift towards their respective endpoints of the body region of the PDF $f_s(S)$. Likewise, if the endpoints of the body region are chosen too far out in the tail, where the probability $f_s(S)$ approaches zero, difficulty will be encountered in applying the Cauchy distribution as a bound. This is easily remedied by limiting the probability that the bound is being fit to some small value greater than zero. Again, this will lead to a loose fit and inefficiencies in the implementation of the acceptance-rejection algorithm. In general, the Cauchy distribution will bound the tail of the distribution of interest loosely. If, however, the tail region is chosen so that the total probability in the tail is small, large inefficiencies in accepting points for the tails will not impact the overall efficiency greatly. Figure 4.3 illustrates the use of the Cauchy distribution as a bound for the right tail of a PDF. Thus, the bound for the PDF $f_s(S)$ is given by equation (4.6) as

$$\alpha f_r(R) = \begin{cases} \dfrac{b_1}{\pi(b_1^2 + (R - \mu_1)^2)} & , R < S_0 \\[2ex] fu_k & , S_0 + (k-1)\Delta S \le R < S_0 + k\Delta S, 1 \le k \le M \\[2ex] \dfrac{b_r}{\pi(b_r^2 + (R - \mu_r)^2)} & , R \ge S_1 \end{cases} \qquad (4.6)$$

The area under the Cauchy bound for the left tail is given by

$$\alpha_1 = \begin{cases} \dfrac{1}{2} + \dfrac{1}{\pi}\tan^{-1}\left(\dfrac{S_0 - \mu_1}{b_1}\right), & \text{left tail exists} \\[2ex] 0 & , \text{otherwise} \end{cases} \qquad (4.7)$$

and the area under the right tail is given by

$$\alpha_r = \begin{cases} \dfrac{1}{2} - \dfrac{1}{\pi}\tan^{-1}\left(\dfrac{S_1 - \mu_r}{b_r}\right), & \text{right tail exists} \\[2ex] 0 & , \text{otherwise} \end{cases} \qquad (4.8)$$

The area at the left edge of each segment, designated as A_k, is given by

$$A_k = \begin{cases} \alpha_1 & , \quad k = 1 \\[1ex] fu_{k-1}\Delta S + A_{k-1} & , \quad 2 \le k \le M+1 \end{cases} \qquad (4.9)$$

and the total area under the bound is found to be

$$\alpha = \alpha_1 + \sum_{k=1}^{M} fu_k \, \Delta S + \alpha_r \qquad . \qquad (4.10)$$

Integrating the bound function and normalizing by the total area yields the cumulative distribution function (CDF) for the bound

$$F_r(R) = \begin{cases} \dfrac{1}{\alpha}(\dfrac{1}{2} + \dfrac{1}{\pi}\tan^{-1}(\dfrac{R-\mu_l}{b_l})) & ,R < S_0, \text{ } \textit{left tail present} \\[4mm] 0 & ,R < S_0, \text{ } \textit{left tail absent} \\[4mm] \dfrac{1}{\alpha}fu_k(R-(k-1)\Delta S - S_0) + A_k & ,S_0 + (k-1)\Delta S \le R < S_0 + k\Delta S \\[4mm] 1 - \dfrac{1}{\alpha}(\dfrac{1}{2} - \dfrac{1}{\pi}\tan^{-1}(\dfrac{R-\mu_r}{b_r})) & ,R \ge S_1, \text{ } \textit{right tail present} \\[4mm] 0 & ,R \ge S_1, \text{ } \textit{right tail absent} \end{cases} \quad (4.11)$$

This can be inverted to yield

$$F_r^{-1}(U) = \begin{cases} b_l\tan(\pi(\alpha U - \dfrac{1}{2})) + \mu_l & ,U < A_1 \\[4mm] \dfrac{\alpha}{fu_k}(U - A_k) + (k-1)\Delta S + S_0 & ,A_k \le U < A_{k+1}, 1 \le k < M \\[4mm] b_r\tan(\pi(\dfrac{1}{2} - \alpha(1-U))) + \mu_r & ,U \ge A_{M+1} \end{cases} \quad (4.12)$$

To generate a variate from f_r, a uniform random sample U is generated. Then

$$R = F_r^{-1}(U) \qquad\qquad (4.13)$$

will have the desired distribution.

As is discussed in section 3, a lower bound can be used to decrease the computational load of the acceptance-rejection algorithm. In the body region of the PDF, a lower bound can be found in the same manner as the upper. An optimization routine is used to find the minima in each region, which are designated fl_k. The lower bound is set to zero in the tail regions. This leads to the bound function

$$h(R) = \begin{cases} fl_k & ,S_0 + (k-1)\Delta S \le S < S_0 + k\Delta S, 1 \le k \le M \\[2mm] 0 & ,\textit{elsewhere} \end{cases} \quad (4.14)$$

This is illustrated in figure 4.4.
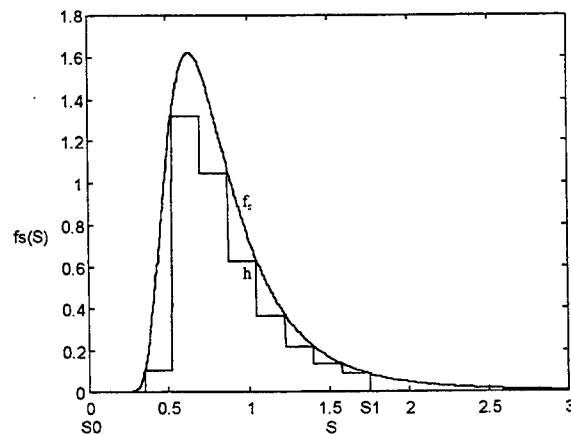
Figure 4.4: Lower Bound of PDF.

5. GENERATION OF RANDOM SAMPLES FOR THE CHARACTERISTIC PDF OF THE STUDENT-T SIRV

The characteristic PDF of the Student-t SIRV is given as

$$f_s(S) = \frac{2b}{\Gamma(v)2^v} b^{2v-1} S^{-(2v+1)} e^{-\frac{b^2}{2S^2}} u(S) \qquad (5.1)$$

with the shape parameter b and the scale parameter υ. The characteristic PDF $f_s(S)$ for $\upsilon=2$ and $b=\sqrt{2}$ is illustrate in figure 5.1. The point separating the body of the PDF from the tail is chosen as $S_1= 1.43$, where the total tail probability is equal to 0.1. Figure 5.2 shows the calculated bounds.
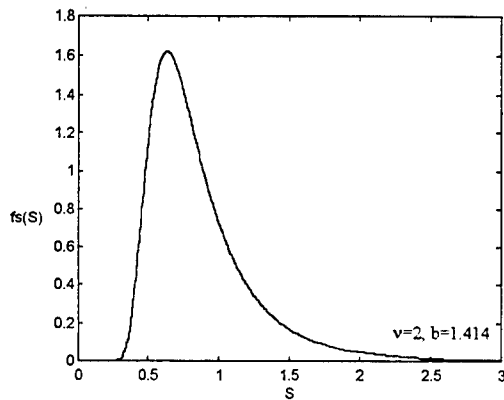
Figure 5.1: Characteristic PDF of the Student-t SIRV.
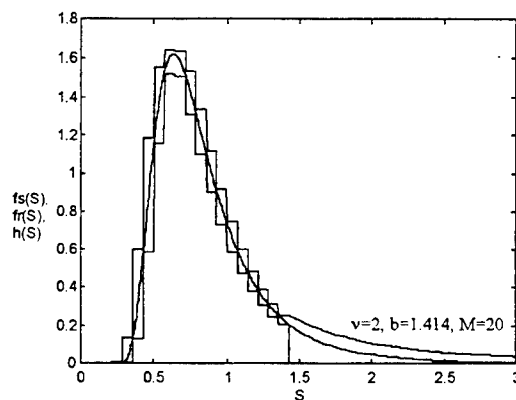
Figure 5.2: Bounds for the Student-t Characteristic PDF.

Figures 5.3 and 5.4 show the calculated CDF function for the bound and the inverse of the CDF, respectively.
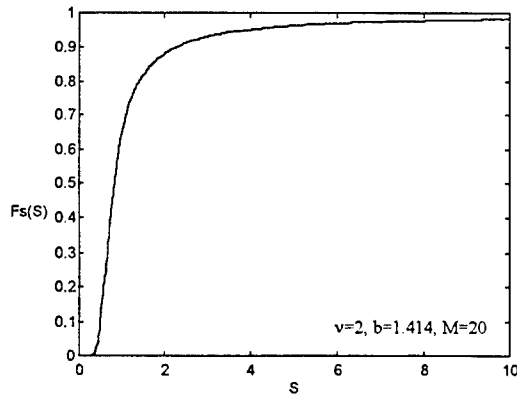
Figure 5.3: CDF of Bound for the Student-t Characteristic PDF.
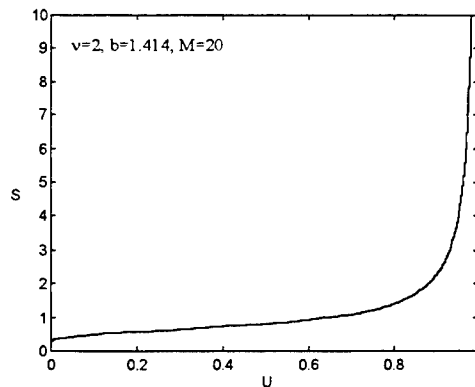
Figure 5.4: Inverse of the Bound CDF.

Figure 5.5 contains a histogram of 10,000 realizations of the random variable S for the characteristic PDF of the Student-t distribution. The efficiency of the generator with respect to the number of points accepted was approximately 78%. In addition, a Kolmogorov-Smirnoff test was used to examine the random data produced by this generator, which was accepted with high significance.
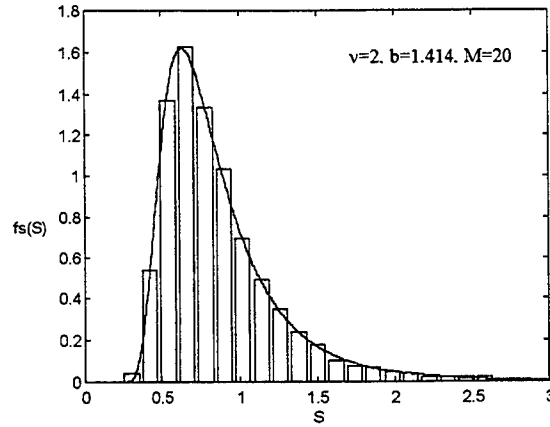
Figure 5.5: Empirical Characteristic PDF for the Student-t SIRV.

6. GENERATION OF SAMPLES FOR THE QUADRATIC FORM OF THE STUDENT-T SIRV

As is discussed in section 2, an SIRV can be expressed in terms of a quadratic form. The distribution for the quadratic form of the Student-t distribution corresponding to the characteristic PDF $f_s(s)$ given in equation (5.1), is given by

$$f_q(Q(Y)) = \frac{b^{2v}\Gamma(v+\frac{N}{2})}{\Gamma(v)\Gamma(\frac{N}{2})}\frac{Q(Y)^{\frac{N}{2}-1}}{(b^2+Q(Y))^{v+\frac{N}{2}}}u(Q(Y)), v > 0. \quad (6.1)$$

In the example, N= 16, υ=2, and b=$\sqrt{2}$. Here the point Q_1 was set to 32.7 for a total tail probability of 0.1. Figures 6.1 and 6.2 show the PDF of the quadratic form and the bounds calculated for it, respectively.
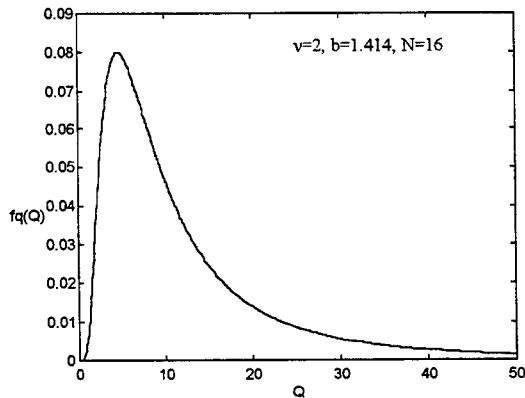
Figure 6.1: PDF of the Student-T Quadratic Form.
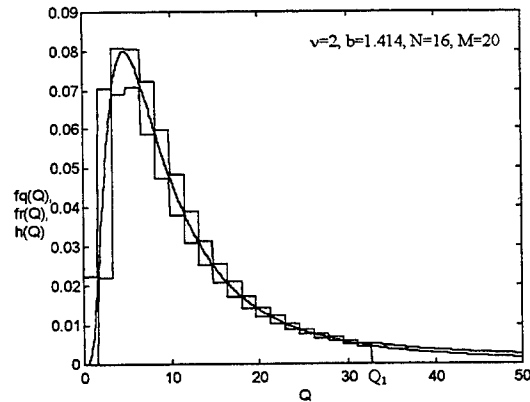
Figure 6.2: Bounds for the Student-t Quadratic Form.

Again, 10000 points were generated for this distribution, and the resulting histogram is shown in figure 6.3. The data generated for the Student-t quadratic form was accepted by the Kolmogorov-Smirnoff test with high significance and the efficiency of the generator was greater than 81%.
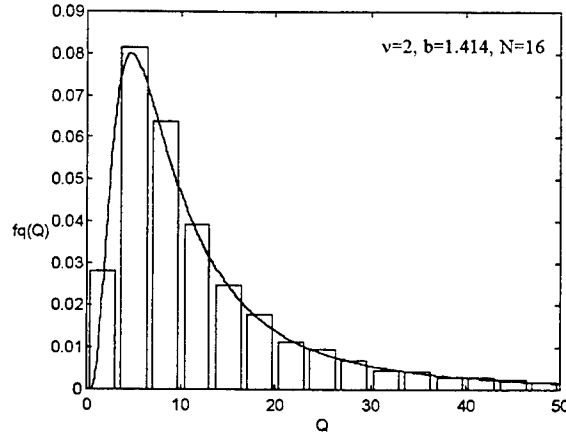
Figure 6.3: Empirical PDF for the Student-t Quadratic Form.

7. THE DISCRETE GAUSSIAN MIXTURE SIRV

The discrete Gaussian mixture SIRV has special significance in that its PDF is a finite weighted sum of Gaussian PDF's. This structure leads to some useful implications with respect to evaluating the Gaussian mixture PDF and the PDF of its quadratic form. It also leads to a simple technique for generating random samples. Through the proper choice of the parameters for the Gaussian mixture, it may be used to approximate many other SIRV's which may not be as readily evaluated.

The discrete Gaussian mixture SIRV has the characteristic PDF,

$$f_s(S) = \sum_{k=1}^{K} w_k \delta(S - s_k), \quad for \ w_k > 0 \quad and \sum_{k=1}^{K} w_k = 1, \qquad (7.1)$$

and the PDF of the discrete Gaussian mixture SIRV is

$$f_y(Y) = (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} h_N(Y), \qquad (7.2)$$

where $h_N(Y)$ is given by

$$h_N(Y) = \sum_{k=1}^{K} w_k s_k^{-N} e^{-\frac{(Y-b)^H \Sigma^{-1}(Y-b)}{2 s_k^2}}. \qquad (7.3)$$

The PDF of the quadratic form of the discrete Gaussian mixture is simply

$$f_q(Q(Y)) = \frac{1}{2^{\frac{N}{2}} \Gamma(\frac{N}{2})} Q(Y)^{\frac{N}{2}-1} \sum_{k=1}^{K} w_k s_k^{-N} e^{-\frac{Q(Y)}{2 s_k^2}}. \qquad (7.4)$$

Consider again the PDF for the quadratic form of the Student-t SIRV given in equation (6.1) with N=16, b=$\sqrt{2}$, and v=2. It can be approximated closely by a discrete Gaussian mixture SIRV with the following ten weights:

Figure 7.1: Gaussian Mixture Weights
Approximating the Student-t SIRV (N=16, v=2, $b = \sqrt{2}$)

w_k	s_k
.0809	.4270
.1024	.5864
.0999	.5614
.0875	.6192
.0800	.7545
.1420	.7937
.1042	.8608
.1116	1.052
.0954	1.201
.0962	1.758

The approximation achieved is shown in figure 7.2, and is a good fit for the body of the distribution. At some point in the tail of the distribution, the term with the largest variance becomes dominant, and the Gaussian mixture behaves like the Gaussian distribution. At this point, the approximation becomes less accurate as the tail of the Gaussian distribution decreases more rapidly than that of other SIRV's, but it is usually at a point far out in the tail, where the total remaining probability is small.

The weights in figure 7.1 are determined by first breaking the characteristic PDF of the Student-t SIRV, given in equation (5.1), into ten equi-probable bins. The variances s_k are determined by calculating the average value of s in each bin. This is shown in figure 7.3. From this starting point, an optimization is performed using a gradient descent algorithm, to minimize the relative error between the PDF of the Student-t quadratic form and the PDF of the Gaussian mixture quadratic form.

Random samples can be generated for the Gaussian mixture SIRV by simply generating a variate from the discrete distribution given in equation (7.1) and multiplying it by a Gaussian random vector. The discrete variate can be obtained by determining which bin an uniformly distributed variate falls into, and choosing the associated value of s_k. Likewise, data can be generated for the quadratic form of the Gaussian mixture by using the selected value of s_k as the variance of a gamma distributed random variable. The gamma distribution is given by

$$f_x(X) = \frac{X^{a-1} e^{-X}}{\Gamma(a)} \; for \; X > 0, \qquad (7.5)$$
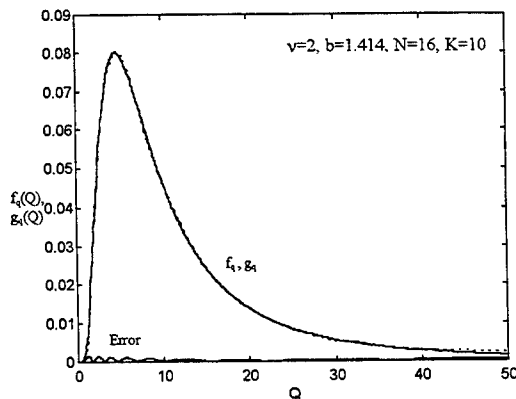
Figure 7.2: Comparison of Student-t Quadratic Form and the Gaussian Mixture Approximation.
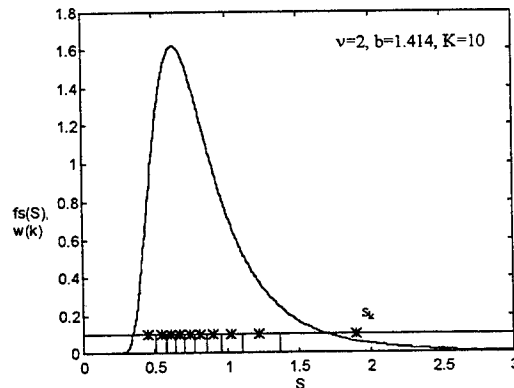
Figure 7.3: Partition of Student-t PDF into Equi-probable Bins.

where a=N/2. The desired random sample is obtained directly from the gamma distributed random variable x according to

$$q(y) = 2s_k{}^2 x \qquad . \qquad (7.6)$$

A histogram of 10,000 random samples for the Gaussian mixture quadratic form specified in figure 7.1 is compared to the PDF of the Student-t quadratic form in figure 7.4.
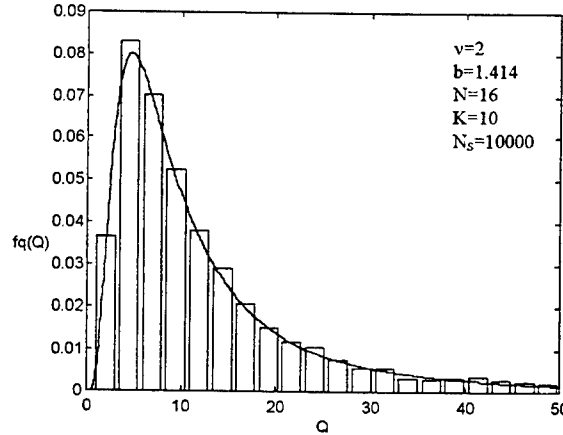
Figure 7.4: Comparison of Student-t Quadratic Form vs. Histogram of Gaussian Mixture Data.

8. PARAMETERIZED NON-GAUSSIAN RECEIVER

Target detection in clutter for a transmitted radar waveform of N coherent pulses is described by the binary hypothesis problem,

$$H_0: \quad \mathbf{r} = \mathbf{d} \qquad \text{(target absent)} \qquad (8.1)$$

$$H_1: \quad \mathbf{r} = a e^{j\phi}\mathbf{s} + \mathbf{d} \qquad \text{(target present)}$$

where the elements of the N-dimensional vectors $\mathbf{r}$, $\mathbf{d}$, and $\mathbf{s}$, are lowpass complex envelope samples of the received data, the disturbance, and the desired signal, respectively. Signal attenuation and target reflection characteristics are modeled by the target amplitude parameter a. The initial phase of the received waveform, which is pulse-to-pulse coherent, is represented by ϕ. D. Stadelman and D. Weiner have shown that the optimal receiver has the form[10]

$$T(\mathbf{r}) = \int_{\Omega_\Phi} \int_{\Omega_A} T(\mathbf{r}|A,\Phi) f_a(A) f_\phi(\Phi) \, dA \, d\Phi \mathop{\gtrless}_{<H_0}^{>H_1} \eta \qquad (8.2)$$

where a and ϕ are assumed to remain constant over a single CPI, and are random variables from CPI to CPI with PDF's $f_a(A)$ and $f_\Phi(\phi)$ respectively. In equation (8.2), the conditional probability $T(\mathbf{r}|A,\Phi)$ is given by

$$T(\mathbf{r}|A,\Phi) = \frac{h_N\left[(\mathbf{r} - a e^{j\Phi}\mathbf{s})^H \Sigma^{-1}(\mathbf{r} - a e^{j\Phi}\mathbf{s})\right]}{h_N(\mathbf{r}^H \Sigma^{-1} \mathbf{r})} \qquad . \qquad (8.3)$$

It has been shown by D.L. Stadelman and D.D. Weiner[10], that whenever the optimum Gaussian test statistic $T_g(\mathbf{r})$ satisfies the condition

$$T_g(c\mathbf{r}) = f(c) T_g(\mathbf{r}), \qquad (8.4)$$

where c does not depend on $\mathbf{r}$ and f(.) is any function, that the receiver has the canonical form of figure 8.1 for any type of SIRV. The condition is satisfied for many commonly used target models, including a U(0,2π) distribution for ϕ and any amplitude PDF, f_a.

(a)   (b)
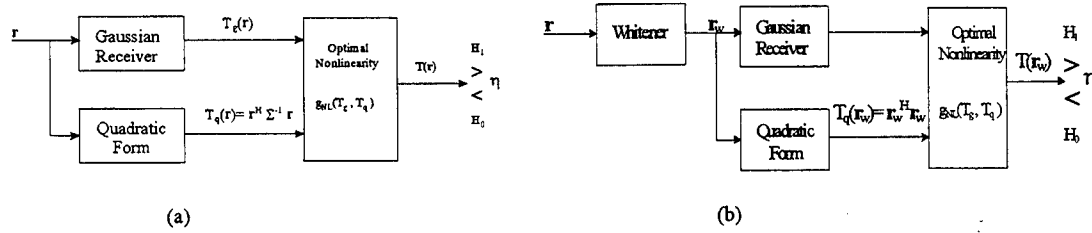
Figure 8.1: Canonical Forms of the Optimal SIRV Processor.

When the clutter can be modeled as a Gaussian mixture SIRV, the receiver specified in equation (8.1), for a U(0,2π) distributed initial random phase and constant amplitude, becomes[4]

$$T_{gm}(\mathbf{r}) = \frac{\sum_{k=1}^{K} w_k s_k^{-N} e^{-\frac{1}{2s_k^2}(\mathbf{r}^H \Sigma^{-1} \mathbf{r} + a^2 \mathbf{s}^H \Sigma^{-1} \mathbf{s})} I_0\left(\frac{a|\mathbf{s}^H \Sigma^{-1} \mathbf{r}|}{s_k^2}\right)}{\sum_{k=1}^{K} w_k s_k^{-N} e^{-\frac{1}{2s_k^2}(\mathbf{r}^H \Sigma^{-1} \mathbf{r})}} . \qquad (8.5)$$

The form of this receiver is illustrated in figure 8.2. This receiver is analogous to a K-channel Gaussian receiver. If the clutter is statistically consistent with the SIRV model, the clutter distribution can be approximated by the Gaussian mixture model and the receiver in figure 8.2 can be used as a sub-optimal approach. If the underlying distribution of the clutter changes, the form of the receiver in figure 8.2 does not change, only the parameters do.

For clutter distributed according to the Student-t SIRV presented in section 6, the optimal receiver for the known amplitude, U(0,2π) random phase signal is found by D.L. Stadelman and D.D. Weiner[10] to be

$$\left(\frac{\left(\mathbf{r}^H \Sigma^{-1} \mathbf{r} + b^2\right)\lambda_r}{\mathbf{r}^H \Sigma^{-1} \mathbf{r} + b^2 + a^2 \mathbf{s}^H \Sigma^{-1} \mathbf{s}}\right)^{v+\frac{N}{2}} P_{v+\frac{N}{2}-1}(\lambda_r) \underset{<_{H_0}}{\overset{>^{H_1}}{}} \eta \quad for \ v > 0, \qquad (8.6)$$

where λ_r is defined as

$$\lambda_r = \left(\frac{\mathbf{r}^H \Sigma^{-1} \mathbf{r} + b^2 + a^2 \mathbf{s}^H \Sigma^{-1} \mathbf{s}}{\sqrt{\left(\mathbf{r}^H \Sigma^{-1} \mathbf{r} + b^2 + a^2 \mathbf{s}^H \Sigma^{-1} \mathbf{s}\right)^2 - \left(2a|\mathbf{s}^H \Sigma^{-1} \mathbf{r}|\right)^2}}\right) \geq 1, \qquad (8.7)$$

and $P_\mu(.)$ is the Legendre function of the first kind of degree μ, which reduces to a Legendre polynomial for integer μ.
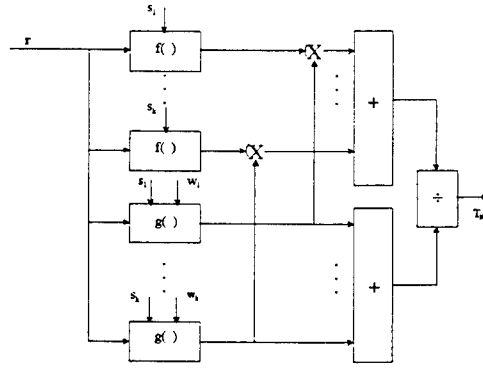
Figure 8.2: Block Diagram of the Optimal Gaussian Mixture Receiver.

The performance of the optimal Student-t receiver is compared to the performance of the optimal Gaussian receiver and the optimal Gaussian mixture receiver for the Gaussian mixture specified in figure 7.1. Figure 8.3 shows the performance of each of the receivers for a probability of false alarm of $P_{fa}=.01$, and with the clutter generated from the Student-t distribution. The signal-to-clutter ratio (SCR) is defined to be

$$SCR = \frac{a^2 \|s\|^2}{E(\|d\|^2)},$$

(8.8)

where d represents the disturbance. While the Gaussian mixture receiver did not perform as well as the optimal Student-t receiver, it significantly outperformed the Gaussian receiver. It is expected that increasing the number of channels K in the Gaussian mixture receiver will improve the performance of the receiver.
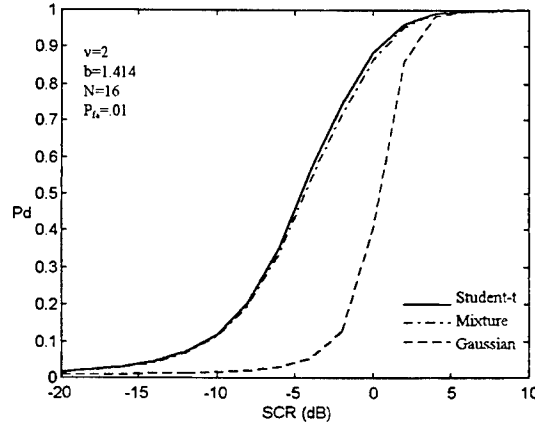
Figure 8.3: Comparison of the Student-t, Gaussian and Gaussian Mixture Receivers.

9. SUMS OF SIRV'S AND THEIR EVALUATION USING THE GAUSSIAN MIXTURE APPROXIMATION

Often, the interference is the result of a combination of several disturbances. Therefore it is desirable to evaluate the PDF of a sum of two SIRV's. Let the received signal be defined as

$$r = y + z$$

(9.1)

where y and z are independent SIRV's with characteristic PDF's $f_s(S)$ and $f_v(V)$ respectively. The PDF of y is obtained from equation (2.5) as

$$f_y(Y) = (2\pi)^{-\frac{N}{2}} \left| \Sigma_y \right|^{-\frac{1}{2}} \int_0^\infty S^{-N} e^{-\frac{(Y-b_y)^H \Sigma_y^{-1}(Y-b_y)}{2S^2}} f_s(S) dS \qquad (9.2)$$

and the PDF of z is likewise given by

$$f_z(Z) = (2\pi)^{-\frac{N}{2}} \left| \Sigma_z \right|^{-\frac{1}{2}} \int_0^\infty V^{-N} e^{-\frac{(Z-b_z)^H \Sigma_z^{-1}(Z-b_z)}{2V^2}} f_v(V) dV. \qquad (9.3)$$

The conditional probabilities for y and z conditioned on s and v are obtained simply from the multivariate Gaussian PDF as

$$f_{y|s,v}(Y|S,V) = (2\pi)^{-\frac{N}{2}} \left| S^2 \Sigma_y \right|^{-\frac{1}{2}} e^{-\frac{(Y-b_y)^H \Sigma_y^{-1}(Y-b_y)}{2S^2}} \qquad (9.4)$$

and

$$f_{z|s,v}(Z|S,V) = (2\pi)^{-\frac{N}{2}} \left| V^2 \Sigma_z \right|^{-\frac{1}{2}} e^{-\frac{(Z-b_z)^H \Sigma_z^{-1}(Z-b_z)}{2V^2}}. \qquad (9.5)$$

Note that y is independent of v, and thus the conditional PDF $f_{y|v}(Y|V)$ is equal to $f_y(Y)$. The same holds true for z with respect to s. The sum r conditioned on s and v is simply the sum of two independent Gaussian random vectors, and the conditional PDF of the sum is known to be multivariate Gaussian with the PDF given by

$$f_{r|s,v}(R|S,V) = (2\pi)^{-\frac{N}{2}} \left| \Sigma_r \right|^{-\frac{1}{2}} e^{-\frac{(R-b_r)^H \Sigma_r^{-1}(R-b_r)}{2}}, \qquad (9.6)$$

with covariance matrix

$$\Sigma_r = S^2 \Sigma_y + V^2 \Sigma_z \qquad (9.7)$$

and mean vector

$$b_r = b_y + b_z. \qquad (9.8)$$

The PDF of the received vector is given by

$$f_r(R) = (2\pi)^{-\frac{N}{2}} \int_0^\infty \int_0^\infty \left| S^2 \Sigma_y + V^2 \Sigma_z \right|^{-\frac{1}{2}} e^{-\frac{(R-b_r)^H \left[S^2 \Sigma_y + V^2 \Sigma_z \right]^{-1}(R-b_r)}{2}} f_s(S) f_v(V) dSdV. \qquad (9.9)$$

In general, the sum r will not be a SIRV unless Σ_z is a scalar multiple of Σ_y. However, if y and z are approximated by Gaussian mixtures, the PDF of r can still be evaluated. From equation (7.1), the characteristic PDF of the Gaussian mixture for each SIRV is given by

$$f_s(S) \approx \sum_{k=1}^K w_k \delta(S - s_k), \quad \textit{for } w_k > 0 \quad \textit{and} \sum_{k=1}^K w_k = 1 \qquad (9.10)$$

and

$$f_v(V) \approx \sum_{k=1}^K p_k \delta(S - v_k), \quad \textit{for } p_k > 0 \quad \textit{and} \sum_{k=1}^K p_k = 1. \qquad (9.11)$$

respectively. Substituting equations (9.10) and (9.11) into equation (9.9) yields

$$f_r(R) \approx (2\pi)^{-\frac{N}{2}} \sum_{k=1}^{K} \sum_{l=1}^{L} w_k \, p_l \left| s_k^2 \Sigma_y + v_l^2 \Sigma_z \right|^{-\frac{1}{2}} e^{-\frac{(R-b_r)^H \left[s_k^2 \Sigma_y + v_l^2 \Sigma_z \right]^{-1} (R-b_r)}{2}} \quad . (9.12)$$

It should be noted that no constant covariance matrix can be defined for all values of s and v in general, and thus a quadratic form does not exist in general.

If one of the SIRV's being summed in equation (9.1) is Gaussian distributed with the PDF

$$f_z(Z) = (2\pi)^{-\frac{N}{2}} \left| \Sigma_z \right|^{-\frac{1}{2}} e^{-\frac{(Z-b_z)^H \Sigma_z^{-1} (Z-b_z)}{2}} \quad , \tag{9.13}$$

then the PDF of the sum reduces to

$$f_r(R) = (2\pi)^{-\frac{N}{2}} \int_0^\infty \left| S^2 \Sigma_y + \Sigma_z \right|^{-\frac{1}{2}} e^{-\frac{(R-b_r)^H \left[S^2 \Sigma_y + \Sigma_z \right]^{-1} (R-b_r)}{2}} f_s(S) \, dS, \tag{9.14}$$

and the approximation using the Gaussian mixture becomes

$$f_r(R) \approx (2\pi)^{-\frac{N}{2}} \sum_{k=1}^{K} w_k \left| s_k^2 \Sigma_y + \Sigma_z \right|^{-\frac{1}{2}} e^{-\frac{(R-b_r)^H \left[s_k^2 \Sigma_y + \Sigma_z \right]^{-1} (R-b_r)}{2}} \quad . (9.15)$$

Equation (9.15) is useful for evaluating the distribution of the disturbance when the disturbance consists of SIRV clutter combined with additive Gaussian noise.

Although the distributions of the sum r given in equations (9.9) and (9.14) do not yield SIRV's, in general, the approximations given in equations (9.12) and (9.15) can be used to model the transitional case where the SIRV clutter is not dominant. For the zero-mean case, when the disturbance can be modeled as a sum of Gaussian mixture SIRV's, the receiver for a U(0,2π) distributed initial random phase and constant amplitude signal becomes[4]

$$T(r) = \frac{\sum_{k=1}^{K} \sum_{l=1}^{L} w_{kl} \left| \Sigma_{kl} \right|^{-\frac{1}{2}} e^{-\frac{r^H \Sigma_{kl}^{-1} r}{2}} e^{-\frac{a^2 s^H \Sigma_{kl}^{-1} s}{2}} I_0 \left(a \left| s^H \Sigma_{kl}^{-1} r \right| \right)}{f_{r|H_0}(R|H_0)}, \tag{9.16}$$

where

$$w_{kl} = w_k \, p_l , \tag{9.17}$$

$$\Sigma_{kl} = s_k^2 \Sigma_y + v_l^2 \Sigma_z , \tag{9.18}$$

and the distribution of the received signal without a target present, given by equation (9.9), reduces to

$$f_{r|H_0}(R|H_0) = \sum_{k=1}^{K} \sum_{l=1}^{L} w_{kl} \left| \Sigma_{kl} \right|^{-\frac{1}{2}} e^{-\frac{r^H \Sigma_{kl}^{-1} r}{2}} . \tag{9.19}$$

Only the case for a signal with constant amplitude has been presented here, but the optimal receivers for signals with fluctuating amplitude can be similarly obtained for the model given in equation (9.9)[4].

10. SUMMARY

The SIRV model is found to be useful for evaluating the non-Gaussian clutter problem. Significant improvement in detection performance has been demonstrated in the literature for the optimal non-Gaussian receiver over conventional receiver designs when the disturbance is SIRV clutter. The use of the SIRV model has led to its own set of problems, however. For simulations, it is necessary that an efficient method of generating SIRV random samples be available. Through the use of a piecewise linear bound, an efficient, with respect to the number of points rejected, acceptance-rejection algorithm can be developed for any sufficiently smooth, bounded distribution. Furthermore, this procedure can easily be automated. The characteristic PDF's of many SIRV's fall within the above category. This can be combined with a Gaussian random number generator to produce multivariate distributions with the desired form and covariance matrix. Furthermore, if the application allows, the quadratic form of the SIRV can be generated directly with this method, avoiding the problems associated with generating independent random vectors.

The Gaussian mixture SIRV can be used to approximate many other SIRV's. This leads to simple approximations for the PDF of a SIRV, and for the PDF of its quadratic form. A simple and highly efficient scheme exists for generating random samples for the Gaussian mixture SIRV, and thus random samples for many other SIRV's can be easily generated using the Gaussian mixture approximation. Furthermore, many other previously unknown SIRV's can be approximated and investigated using the Gaussian mixture SIRV, and the Gaussian mixture SIRV leads to a sub-optimal parameterized receiver which has the same form for all SIRV's.

11. ACKNOWLEDGMENTS

12. REFERENCES

[1]   K. Yao, "A Representation Theorem and Its Applications to Spherically Invariant Random Processes", IEEE Transactions on Information Theory, vol IT-19, pp. 600-608, 1973.

[2]   M. Rangaswamy, D. Weiner, and A. Ozturk, "Non-Gaussian Random Vector Identification Using Spherically Invariant Random Process", IEEE Transactions on Aerospace and Electronic Systems, vol. 29, pp 111-124, January, 1993.

[3]   M. Rangaswamy, D. Weiner, and A. Ozturk, "Computer Generation of Correlated Non-Gaussian Radar Clutter", IEEE Transactions on Aerospace and Electronic Systems, vol. 31, pp. 106-116, January, 1995.

[4]   D. Stadelman, Ph.D. dissertation in progress, Syracuse University, Syracuse, NY, 1995.

[5]   D.E. Knuth, The Art of Computer Programming Vol. 2: Seminumerical Methods, Addison-Wesley, Reading, MA, 1981.

[6]   Paul Bratley, Bennett L. Fox, and Linus E. Schrage, A Guide to Simulation, Springer-Verlag, New York, NY, 1987.

[7]   A. Keckler, Ph.D. dissertation in progress, Syracuse University, Syracuse, NY, 1995.

[8]   W.H. Press, et. al., Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, New York, NY, 1992.

[9]   M.E. Johnson, Multivariate Statistical Simulation, John Wiley and Sons, New York, NY, 1987.

[10]   D.L. Stadelman and D. D. Weiner, "Detection of Weak Signals with Random Parameters in Non-Gaussian Clutter",1995 SPIE International Symposium on Optical Science, Engineering, and Instrumentation, San Diego, CA, July, 1995.

Associate did not participate in program.

AUTOMATED CREATION OF WEB FORMS FOR CREST DATA LIBRARY

Brian Lubinsky
Graduate Student
Department of Electrical Engineering

University of Cincinnati
ML -0152
Cincinnati. OH 45220

Final Report for:
Graduate Student Research Program
Rome Laboratory

7-1

AUTOMATED CREATION OF WEB FORMS FOR CREST DATA LIBRARY

Brian D. Lubinsky
Graduate Student
Department of Electrical Engineering
University of Cincinnati

Abstract

A method of automatically creating web forms (html files) through the use of a GUI (graphical user interface) was created. Tcl/tk programming language was used. The GUI prompts the user for a variety of information, and after completing the data entry, the user may simply click on a button icon to create an html document. The user may then view this or any other form through a web browser or may choose to simply view the text file through an editor. This method provides a quick and easy way of creating web documents, regardless of previous internet or html experience. Help windows are also a prominent feature of the program.

AUTOMATED CREATION OF WEB FORMS FOR CREST DATA LIBRARY

Brian D. Lubinsky

Introduction

The CREST (Common Research Environment for STAP Technology) data library will allow the user to query and retrieve data from the Mountain Top Summit Database. Data will be the results of experiments carried out under RLSTAP (Rome Labs Space Time Adaptive Signal Processing), a simulation tool designed in order to simulate radar systems before prototyping hardware. RLSTAP experiments will typically be used to develop tracking algorithms, test clutter models, and model jammers. In order to make experiment results available through the data library, a file compatible for use with the WWW must be created. The program I have written allows for easy implementation of this task through use of a self explanatory graphical user interface, and will automatically create an html (the language used for web files) file from the data the user inputs.

Discussion of Problem

The first consideration in designing the program was to decide what information was necessary to build a complete CREST data form. It was decided that the user must be prompted for an experiment name (for the html file to be saved under), name of the source data file, named of the saved RLSTAP procedure, names of snapshots of any workspaces and output plots related to the experiment, name of the header file created by RLSTAP, name of a comment file contaning notes on the experiment, and, of course, the name of

the user. The program should also be able to provide a listing of available html documents and allow the user to easily access any of these through an editor or a WWW browser.

The program should automatically create an html file after the user is done entering data. When viewing the html file through a web browser, all information given by the user will be displayed and links to experiment comments, header file, workspaces, and output plots are provided (meaning that any of these may be accessed simply by clicking on the appropriate highlighted area of text.)

In order to make the program as widely accessible as possible, help screens explaning its use and fuctionality are provided.

Methodology

The Tcl/tk script writing language was used to create a menuing system. The main menu contains the options build form, view form, help, and quit. Clicking on any of these options will bring the corresronding sub-menu.

The build form options will bring up a sub-menu containing the options "Select username" and "Build a form". Clicking on "Select username" will bring up another sub-menu containing a list of users (composed of engineers and scientists at Rome Labs). Selecting a name from this list will set the user name, phone, and e-mail address. If "other" is chosen, a window prompting the user for his name, phone, and e-mail will appear. Clicking on "Build a form" will bring up a window containing prompts and entry spaces for the information discussed in the problem explanation. Prompts for entering

7-4

workspaces and plots ask for the number of snapshots of each. After entering a number and hitting return, a window will come up in which the user may enter names for the appropriate number of snapshots. Also available in the "build a form" window is an optional change directory entry and a help button which will bring up a window explaining use of the "build form" entry window. After selecting a username and completing the entries, the user may click on a button labeled "Create web page" and the data will be written into two html files whose name is given by the user selected experiment name. One of these is simply a comment file and is created by reading the contents of the comment file provided by the user and writing the text back into an html coded file whose name will be that of the original comment file plus a .html extension. A link to this file is provided by the main html file which is given the name selected by the user (experiment name). This form lists all pertinent entries made by the user and provides the aforementioned link to the comment file as well as links to the header file and snapshots of the RLSTAP workspaces and output plots.

Selecting the "view form" option from the main menu will bring up a window containing a listing of all web forms in the current directory (all files with the .html extention). The listing also contains information on dates of file creation, file size, read/write protection, and ownership. This was done by making a system call to UNIX from the Tcl program and using a ls -al command, then reading the results into a Tcl variable. The user may then select from this list, or another directory if file names are

known, a form to be viewed by entering the file name into the appropriate entry box. After this is done, the user may view the file either as text or through a WWW browser by clicking on buttons labeled "vi", "textedit" (for text viewing) or "mosaic" and "netscape" (web browsers). This is also done by making a UNIX system call through Tcl and will bring up new windows containing the desired information.

The "help" option of the main menu contains four options labeled "What is Post?", "How to build a form to be posted", "Version history", and "Information on CREST" which will bring up windows containing the corresponding information.

Results

Functionality was fully implemented but there were several small flaws. One problem was that I was unable to create a window from inside a window without destroying the button or entry that called the sub-window. A result of this is that the help option from inside the "Build Form" window may only be called once. I compensated for this by putting a warning informing the user of this problem in the help window. Anothe problem occurred in the "View Form" window. A change directory option was included but I was unable to simultaneously update the listing of html files. In order to search new directory, the user must change directories, close the "View Form" window, then re-open it.

Aside from these two bugs, everything worked fully as expected and was quick and easy to use.

Conclusion

The GUI method of implementation provides several advantages over other potential methods, the foremost among these being its simplicity of use. The user is able to easily navigate between menus and may easily update forms or correct misentered information. Development of this tool will continue in the future an allow the user to Post the web forms to the CREST data library.

References

1. Tcl and the Tk Toolkit

   John K. Ousterhout

2. HTML Documentation and Style Guide

   Ian S. Graham

Addendum: Other suport efforts performed

Another task I performed was verification of Math Tools for RLSTAP. Math tools are a set of glyph functions available through the RLSTAP KHOROS workspace, including everything from basic functions such as addition and subtraction to more advanced matrix manipulation to signal processing tools such as FFT algorithms. I was able to discover and publicize several problems, which eventually led to changes in the source code. One problem was the user entry format for zerofill, a glyph padding a matrix with zeros, which I felt was confusing and would lead to hard to detect errors. Labeling clarity for this and several other glyphs were improved over the summer. I also ran some RLSTAP lineups and compared results with MATLAB only to discover that the 2-D FFT was not functioning correctly. This has been corrected as well. Aside from these problems which have been ironed out, I found the system performed very well and was able to handle very large matrices (up to about 1000x1000) and was accurate over a numerical range of roughly 1e-40 to 3e+38.

```
#!/usr/local/bin.sun4/wish -f
frame .menu -relief raised -borderwidth 1
message .msg -relief raised -borderwidth 1 -text "Main Menu\n"
label .bitmap -bitmap @/n/mountain/users/user1/CREST.xbm

pack .msg -side top -fill x
pack .bitmap .menu -side bottom -expand yes -fill both

# global variables accessed in subroutines
set nm 0
set phone 0
set mail 0


# menus  .menu
#           .build.m
#                .build    Calls up window in proc name
#           .view.m
#           .post.m
#           .help.m    Calls up windows in procedures helpone, helptwo
#                             helpthree, and helpfour


# Global variables  plott() : array of plot names
#                 snap() : array of lineup names

menubutton .menu.build -text "Build form " -menu .menu.build.m
menu .menu.build.m
.menu.build.m add command  -label "Build a form" -command {build $nm}
.menu.build.m add cascade -label "Select username" -menu .menu.build.m.name


menu .menu.build.m.name
.menu.build.m.name add command -label "D. Fabozzi" -command \
{set nm "D. Fabozzi"}
.menu.build.m.name add command -label "M. Callahan" -command {set nm \
"M. Callahan"}
.menu.build.m.name add command -label "L. Adzima" -command \
{set nm "L. Adzima"}
.menu.build.m.name add command -label "M. Pugh" -command {set nm "M. Pugh"}
.menu.build.m.name add command -label "R. Wilson" -command \
 {set nm "R. Wilson"}
.menu.build.m.name add command -label "S. Kelly" -command {set nm "S. Kelly"}
.menu.build.m.name add command -label "J. Carlo" -command {set nm "J. Carlo"}
.menu.build.m.name add command -label "S. Scott" -command {set nm "S. Scott"}
.menu.build.m.name add command -label "E. Starczewski" -command \
```

```tcl
{set nm "E. Starczewski"}
.menu.build.m.name add command -label "P. Zulch" -command {set nm "P. Zulch"}
.menu.build.m.name add command -label "L. Flanders" -command \
{set nm "L. Flanders"}

.menu.build.m.name add command -label "J. Michels" -command \
{set nm "J. Michels"}
.menu.build.m.name add command -label "B. Himed" -command {set nm "B. Himed"}
.menu.build.m.name add command -label "Other" -command {select}

proc select {{s .b9}} {
    catch {destroy $s}
    toplevel $s
    wm title $s "Username entry"
    label $s.l1 -text "Enter username:"
    entry $s.nm -width 20 -relief sunken -textvariable nm
    label $s.l2 -text "Enter phone number:"
    entry $s.phone -width 20 -relief sunken -textvariable phone
    label $s.l3 -text "Enter e-mail address:"
    entry $s.mail -width 20 -relief sunken -textvariable mail
    button $s.b9 -text OK -width 10 -command "destroy $s"
    pack $s.l1 $s.nm $s.l2 $s.phone $s.l3 $s.mail $s.b9 -side top
}




menubutton .menu.view -text "View"  -menu .menu.view.m
menu .menu.view.m
.menu.view.m add command -label "List and view forms in current directory " \
-command ls


# Will list all forms with .html extension and allows directory change
proc ls {{w .b1}} {                                        catch {destroy
$w}
    toplevel $w
    wm title $w "View form"
        set x [glob  *.html -nocomplain]
        set y [llength $x]

        label $w.m1 -text "Listing of forms in [pwd]\n"
        pack $w.m1 -side top
        for {set a 1} {$a < $y} {incr a +1} {
```

```
        set formnm [lindex $x $a]
        label $w.m($a) -text "$formnm"
        pack $w.m($a) -side top
    }
    label $w.l1 -text "\nTo change directories, enter new directory below \
     and <Return>"
    entry $w.dir -width 30 -relief sunken -textvariable dir
    label $w.l2 -text "Enter name of the form to be displayed."
    entry $w.form -width 30 -relief sunken -textvariable form
    label $w.l5 -text "\nWarning: Netscape may not run well on networked terminals"
     label $w.sp -text "\n"
    frame $w.f1
    frame $w.f2
    pack $w.l1 $w.dir $w.l2 $w.form $w.sp $w.f1 $w.sp $w.f2 $w.sp -side top

    label $w.f1.l3 -text "\nText file display options:     "
    label $w.f2.l4 -text "\nBrowser display options :      "
    button $w.f1.b2 -text "View only" -width 10 -command {exec xterm -e \
more $form &}
    button $w.f1.b3 -text "Editor" -width 10 -command {exec textedit \ $form &}
    button $w.f2.b4 -text "Mosaic" -width 10 -command {exec mosaic $form &}
    button $w.f2.b5 -text "Netscape" -width 10 -command {exec netscape \ $form &}

    button $w.b1 -text OK -width 10 -command "destroy $w"
    pack $w.l5 $w.b1 -side bottom
    bind $w.dir <Return> {cd $dir}

    pack $w.f1.l3 $w.f1.b2 $w.f1.b3 -side left
    pack $w.f2.l4 $w.f2.b4 $w.f2.b5 -side left
}




menubutton .menu.post -text "Submit" -menu .menu.post.m
menu .menu.post.m
.menu.post.m add command -label "Build a form" -command exit
.menu.post.m add command -label "View a form" -command exit




menubutton .menu.help -text "Help " -menu .menu.help.m
menu .menu.help.m
.menu.help.m add command -label "What is Post?" -command helpone
.menu.help.m add command -label "How to build a form to be posted" \
```

```
-command helptwo
.menu.help.m add command -label "Version history" -command helpthree
.menu.help.m add command -label "Information on CREST" -command helpfour


menubutton .menu.quit -text "Quit" -menu .menu.quit.m
menu .menu.quit.m
.menu.quit.m add command -label "Quit" -command exit

proc build {n {w .b1}} {
# variables nm,exp,source,procname,lineups,plots,header,comment,phone, mail,dir
    global nm phone mail
    set nm $n
    catch {destroy $w}
    toplevel $w
    dpos $w
    wm title $w "Build form"
    wm iconname $w "tcl sucks"


frame $w.f1
frame $w.f2
frame $w.f3
frame $w.f4
frame $w.f5
frame $w.f6
frame $w.f7
frame $w.f8
frame $w.f9
frame $w.f10


label $w.l1 -text "Author: $nm"
label $w.f1.q1 -text "Experiment name:"
entry $w.f1.exp -relief sunken -textvariable exp
label $w.f2.q2 -text "Enter the source data file (n = none)"
entry $w.f2.source -relief sunken -textvariable source
label $w.f3.q3 -text "Enter the saved procedure"
entry $w.f3.procname -relief sunken -textvariable procname
label $w.f4.q4 -text "Enter the number of saved workspaces"
entry $w.f4.lineups -relief sunken -textvariable lineups
label $w.f5.q5 -text "Enter the number of associated plots"
entry $w.f5.plots -relief sunken -textvariable plots
label $w.f6.q6 -text "Enter the headerfile produced from lineup."
entry $w.f6.header -relief sunken -textvariable header
```

```
label $w.f7.q7 -text "Enter the comment file produced from lineup"
entry $w.f7.comment -relief sunken -textvariable comment
label $w.f8.q8 -text "Optional: enter new directory and <Return>"
entry $w.f8.dir -textvariable dir
button $w.f9.b3 -text "Click here for help" -command "bldhlp $w"
button $w.f10.b1 -text OK -command "destroy $w"


pack  $w.f1 $w.f2 $w.f3 $w.f4 $w.f5 $w.f6 $w.f7 $w.f8 $w.f9 $w.f10 -side top
pack $w.f1.q1  -side left
pack $w.f1.exp -side right
pack $w.f2.q2
pack $w.f2.source -side right
pack $w.f3.q3
pack $w.f3.procname -side right
pack $w.f4.q4 $w.f4.lineups -side left
pack $w.f5.q5 $w.f5.plots -side left
pack $w.f6.q6 $w.f6.header -side left
pack $w.f7.q7 $w.f7.comment -side left
pack $w.f8.q8 $w.f8.dir -side left
pack $w.f9.b3 -side left
pack $w.f10.b1 -side left




bind $w.f4.lineups <Return> {lineup $lineups}
bind $w.f5.plots <Return> {plot $plots}
bind $w.f7.comment <Return> {textfile $comment}
bind $w.f6.header <Return> {textfile $header}
bind $w.f8.dir <Return> {cd $dir}

# This sets phone and mail according to username
switch $nm {
    "D. Fabozzi" {set phone x7750 ; set mail fabozzi@moses.oc.rl.af.mil}
    "M. Callahan" {set phone x7373 ;set mail callahanm@lonexb.admin.rl.af.mil}
    "L. Adzima" {set phone (203)268-1249; set mail larry@tsc.com}
    "M. Pugh" {set phone x7445 ;set mail pugh@mountain.oc.rl.af.mil}
    "R. Wilson" {set phone (203)268-1249 ;set mail randy@tsc.com}
    "S. Kelly" {set phone x7093 ;set mail skelly@mountain.oc.rl.af.mil}
    "J. Carlo" {set phone x7839; set mail carloj@lonexb.admin.rl.af.mil}
    "S. Scott" {set phone x4431;set mail scotts@lonexb.admin.rl.af.mil}
    "E. Starczewski" {set phone x7127;set mail stare@edstar}
    "P. Zulch" {set phone x7861;set mail zulchp@lonexb.admin.rl.af.mil}
```

```
    "L. Flanders" {set phone x7811;set mail flandersl@lonexa.admin.rl.af.mil}
    "J. Michels" {set phone x4431;set mail michelsj@lonexa.admin.rl.af.mil}
    "B. Himed" {set phone x4431;set mail michelsj@lonexa.admin.rl.af.mil}
}

frame $w.fl1
pack $w.fl1 -side bottom
button $w.fl1.b2 -text "Click here to create web page." \
 -command {webpage $nm $exp $phone $mail $comment $header $source $procname \
$lineups $plots plott snap}
pack $w.fl1.b2 -side left
}


# procedure called by help button in build form menu
proc bldhlp {z {z.b3 .b1}} {

    catch "destroy $z.b3"
    toplevel $z.b3
    wm title $z "Build help "
    message $z.b3.l1 -aspect 300 -text "Experiment name: This will be the title of your
document. Document will be saved as (exp name).html
\nSource data file: The name of the file which contains the simulated or recorded data.
\nSaved procedure: File which experiment was saved under in RLSTAP.
\nNumber of workspace lineups: Saved RLSTAP workspaces. Hit <Return> to bring up
window to enter lineup names.
\nNumber of snapped plots: Saved RLSTAP plots. Hit <Return> to bring up window to
enter plot names.
\nDirectory change: Enables user to change directories through TCL. No indication will be
given after the directory has been changed.
\nCreate web page: After selecting a user name and completing the form, click to create
an html file.  No indication will be given after the file is created.
\n\nWarning: After closing this help window, the \"Help\" button will be destroyed."
    button $z.b3.b1 -text OK -width 10 -command "destroy $z.b3"
    pack $z.b3.l1   -side top -fill both
    pack $z.b3.b1 -side bottom
}


# This procedure takes information entered  and creates a web page
proc webpage {nm exp phone mail comment header source procname lineups plots \
pl sn} {
```

```
upvar $pl plottt
upvar $sn snapp

set f [open $exp.html w]
puts $f "<HR>"
puts $f "<TITLE> $exp </TITLE>"
puts $f "<H1> Presentation document for : $exp </H1>"
puts $f "<HR>"
puts $f "<H2> PROCEDURE NAME: $exp"
puts $f "<BR>"
puts $f "<h2>AUTHOR: $nm</h2>"
puts $f "<BR>"
puts $f "<h2>PHONE: $phone</h2>"
puts $f "<BR>"
puts $f "<h2>EMAIL: $mail</h2>"
puts $f "<BR>"
puts $f "<h2>datafile: $source </h2><br>"
puts $f "<h2>Saved procedure location: $procname </h2>"
puts $f "<h2> The <a href=\"$comment.html\"> experiment summary</a> </h2>"
puts $f "<h2> The <a href=\"$header\"> header file</a> </h2> <br>"
for {set i 1} {$i <= $lineups} {incr i +1} \
    {puts $f "<h2>workspace snapshot: <a href=\"$snapp($i)\">$snapp($i)</a>\   </h2>
<br>"}
for {set i 1} {$i <= $plots} {incr i +1} \
    {puts $f "<h2>output plot: <a href=\"$plottt($i)\">$plottt($i)</a> </h2>\ <br>"}
close $f

set g [open $comment.html w]
set h [open $comment r]
puts $g "<TITLE>$comment</TITLE>"
puts $g "<h2>Program Comments for $exp</h2>"
puts $g "<HR>"
while {[gets $h a] != -1} {puts $g "<h3>$a</h3>"}
close $g
close $h
return
}



# A procedure to list texfiles called by proc name:not used
proc textfile {ls {zz .b4}} {
    catch {destroy $zz}
```

```tcl
        toplevel $zz
        if { $ls != "1" } { destroy $zz}
        label $zz.list -text "Text files in current directory [pwd] :\
        \n [glob *.*]"
        button $zz.b4 -text OK -command "destroy $zz"
        pack $zz.list $zz.b4
}


# This is called by proc name and allows entry of file names for plots
# or gives a listing of snapshots
# internal variables   p:Number of plot files
#                 $z.plott($i) : Array of file names of plots
proc plot {p {z .b3}} {
    catch {destroy $z}
    toplevel $z
    wm title $z "Plot name entry"
    if {$p == "1"} {
        label $z.list -text "Snaps in current directory [pwd]:\              \n [glob *.rs]"
        pack $z.list
    }
    if {$p != "1"} {
        for {set i 1} {$i <= $p} {incr i +1} {
        label $z.q($i) -text "Enter file name (snap) for plot #$i"
        entry $z.plott($i) -width 30 -relief sunken -textvariable plott($i)
        pack $z.q($i) $z.plott($i)
        }

}
button $z.b3 -text OK -width 10 -command "destroy $z"
pack $z.b3 -side bottom
}


# This is called by proc name and allows entry of file names for lineups
# or gives a listing of snapshots
# internal variables  a:number of lineups
#                 $y.snap : array of file names of lineups
proc lineup {a {y .b2}} {
    catch {destroy $y}
    toplevel $y
    wm title $y "Lineup name entry"
    if {$a == "1"} {
```

```
        label $y.list -text "Snaps in current directory [pwd]:\          \n [glob *.rs]"
        pack $y.list
    }
    if {$a != "1"} {
    for {set i 1} {$i <= $a} {incr i +1} {
        label $y.q($i) -text "Enter file name (snap) for lineup #$i."
        entry $y.snap($i) -width 30 -relief sunken -textvariable snap($i)
        pack $y.q($i) $y.snap($i)
    }
    }
button $y.b2 -text OK -width 10 -command "destroy $y"
pack $y.b2 -side bottom
}


proc helpone {{w .b1}} {
    catch {destroy $w}
    toplevel $w
    wm title $w "What is post?"
    message $w.m -aspect 300 -text "Post is a program which allows the user \
to document an experiment in the World Wide Web (WWW) form.  This program \
further provides the capability to view the form and subsequently submit \
the document to the CREST library."
pack $w.m -side top -fill both
button $w.b1 -text OK -width 10 -command "destroy $w"
pack $w.b1 -side top -expand yes -pady 2
}


proc helptwo {{w .b1}} {
    catch {destroy $w}
    toplevel $w
    wm title $w "How to build a form"
    message $w.m -aspect 300 -text "To build a form:\n   1. You need to have \
already saved sun-snapshot images of cantata workspaces or plots. \
\n   2. You need to have already saved the header file and a comment file \
(saved with *.txt extensions).\n   3. The WWW browser (in this case Netscape)\ is in
your path.\n \n   The program will lead you through a series of \
questions in which you will enter the filenames of corresponding saved \
snapshots and text files.  The program operates out of a local working \
directory wher you keep these files.  The program then allows you to view the \
form through a WWW browser or display on screen. Finally, you can submit this \
form to the CREST library once the form is complete."
```

```
pack $w.m -side top -fill both
button $w.b1 -text ok -width 10 -command "destroy $w"
pack $w.b1 -side top -expand yes -pady 2
}


proc helpthree {{w .b1}} {
    catch {destroy $w}
    toplevel $w
    wm title $w "Version history"
    message $w.m -aspect 300 -text "#v1.0;\n#      1. beta version\
\n#     2. basic functionality\n#     3. command line interface\
\n#     4. buggy\n\n# v1.1;\n#      1.minimize typing, listing of files\
\n#     2. online help\n#     3. kid proof\n#     4. mail facility working \
\n#     5. added J.Michels, B. Himed to database\n\n# v1.2;\n#      \
1. browser selection\n#     2. fix file not found error for linesnap\
\n#     3. added capability to input multiple workspace lineup snaps\
\n#     4. checks for procedure name and datafile name and copies these\
\n#     files to the baseline to global directories\n#     5. Developed help \
facility a bit"
pack $w.m -side top -fill both
button $w.b1 -text ok -width 10 -command "destroy $w"
pack $w.b1 -side top -expand yes -pady 2
}


proc helpfour {{w .b1}} {
    catch {destroy $w}
    toplevel $w
    wm title $w "Information on CREST"
    message $w.m -aspect 300 -text " The CREST (Common Research Environment \
for Space-Time Processing) program provides a common set of tools and data to \
expedite the research process. The main tenets are: \
\n           1. The RLSTAP/ADT\
\n           2. The CREST Data Library\
\n           3. The Maui High Performance Computing Center \
\n\n For more information about CREST see the home pages :\
\n\n      http://www.mhpcc.edu/otherpages/mountaintop/mtntop.html \
\n      http://www.rl.af.mil:8001/Technology/Demos/STAP/RLSTAP.html"
pack $w.m -side top -fill both
button $w.b1 -text OK -width 10 -command "destroy $w"
pack $w.b1 -side top -expand yes -pady 2
}
```
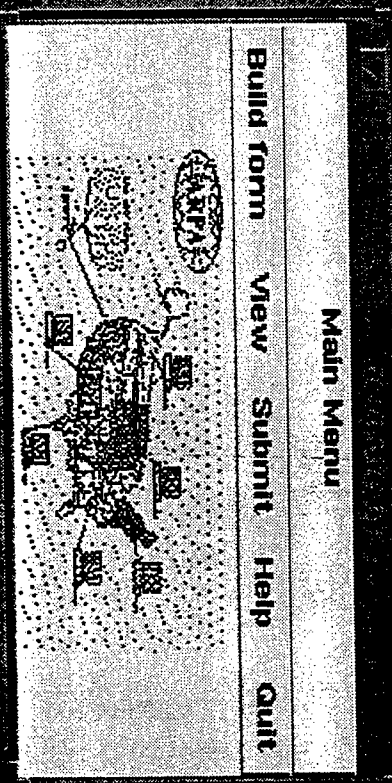
To change directories, enter new directory below and «Return»

Listing of forms in /h/mountain/users/user1

-rw-rw-r-- 1 fabozzi 515 Aug 17 11:27 #ill.html#
-rw-r--r-- 1 user1 88 Aug 11 10:02 .html
-rwxrwxr-x 1 fabozzi 550 Aug 15 14:09 ill.html%

Enter name of the form to be displayed.

Text file display options:

vi   (Sun) Textedit

Browser display options :

Mosaic   Netscape

OK

Warning: Netscape may not run well on networked terminals.

Main Menu

Build Form   View   Submit   Help   Quit

7-19

Build form

Author: 0
Experiment name:

Enter the source data file (n = none)

Enter the saved procedure

Enter the number of saved workspaces

Enter the number of associated plots

Enter the headerfile produced from lineup.

Enter the comment file produced from lineup
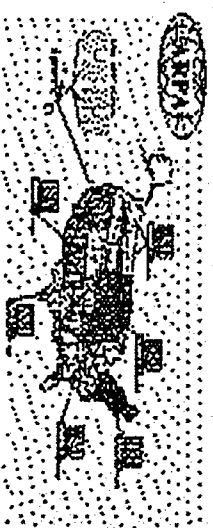
Optional: enter new directory and <Return>

Click here for help

Click here to create web page.

OK

Main Menu

Build form   View   Submit   Help   Quit

PACKAGING AND TESTING OF HIGH SPEED
VERTICAL CAVITY SURFACE EMITTING LASERS
FOR VERTICAL OPTICAL INTERCONNECTS

Sean S. O'Keefe
Graduate Research Assistant
Department of Electrical Engineering

Cornell University
426 Phillips Hall
Ithaca, NY 14853-5401

8-1

PACKAGING AND TESTING OF HIGH SPEED VERTICAL CAVITY SURFACE EMITTING LASERS FOR VERTICAL OPTICAL INTERCONNECTS

Sean S. O'Keefe
Graduate Research Assistant
Department of Electrical Engineering
Cornell University

Abstract

Vertical cavity surface emitting lasers (VCSEL's) are important sources for optical communication architectures, but they have the problem of being very temperature sensitive. Thus, heat sinks were fabricated on metallized thin film diamond wafers and VCSEL's were flip-chip bonded to these heat sinks. The packaged devices were then tested using microwave modulation techniques to determine their -3dB modulation bandwidth. Devices with ten and fifteen micron diameters were tested and the 15 μm diameter devices showed a maximum bandwidth of 6.6 GHz. These devices and this bonding sequence are compatible with continuing work on the vertical optical interconnect project at Rome Laboratory.

PACKAGING AND TESTING OF HIGH SPEED
VERTICAL CAVITY SURFACE EMITTING LASERS
FOR VERTICAL OPTICAL INTERCONNECTS

Sean S. O'Keefe

Introduction

Vertical cavity surface emitting lasers (VCSEL's) offer many advantages to

the field of communications, such as high bandwidths at low currents, single

mode output, and vertical emission[1,2,3]. Like any other laser, a VCSEL consists

of mirrors that form a cavity and a gain medium inside that cavity. Unlike the

typical edge-emitting laser, though, VCSEL's have a one wavelength long cavity

sandwiched between the two in-plane mirrors. The mirrors are Bragg reflectors

and are typically grown by molecular beam epitaxy (MBE) as alternating layers of

GaAs and AlGaAs. A thick enough stack of alternating high and low dielectric

constant materials makes a mirror with greater than 99.9% reflectivity over a

narrow band of wavelengths. Grown next are quantum wells centered in a one

wavelength long thickness of GaAs to form the gain region of the laser. On top of

this, another Bragg reflector is grown. The narrowness of both the Bragg reflector

stop band and the gain spectrum of the quantum wells, and the very wide mode

spacing gives the device a single mode output Thus, without any special processing to form gratings, such as electron-beam lithography or holography, single mode output is achieved. Also, since the output beam is perpendicular to the wafer and the aperture size and positions are defined by lithography, custom tailoring of arrays of devices is possible.

These advantages do not come without problems though. The MBE growth of these devices is very challenging and because of the thickness of the Bragg reflectors, resistive heating of the devices affects performance[4,5,6]. The lasing wavelength is fixed by the length of the cavity, but that length changes as a function of temperature. Also, the gain spectrum shifts wavelength at a different rate as a function of temperature. Therefore, for a particular design wavelength, there is an optimum operating temperature. Similarly, if devices can be held at a known temperature, then the lasing wavelength, threshold current, and output power will be stable and repeatable.

The packaged lasers are to be used as emitters in a vertical optical interconnect (VOI) design at Rome Laboratory[7]. The current design uses a 4x4 array of LED's (660nm) on 400µm centers as transmitters and silicon metal-semiconductor-metal photodetectors as receivers in a parallel high speed optical bus to communicate between different levels of a stacked multi-chip module. If

8-4

the packaging of the VCSEL's is shown to be reliable and the devices show good microwave performance, then the next phase of the project will be to incorporate them into the interconnect architecture. In this report the fabrication of thin film diamond heat sinks for a set of previously fabricated VCSEL's is discussed and the techniques used to bond the devices are outlined. Finally, the optical setup for testing the microwave response of the devices is described and the data is presented.

Fabrication of the diamond heat sinks

Diamond is a nearly perfect electrical insulator (> 11 MΩ-cm)[8] and the best thermal conductor (up to 20 W/cm-K) commercially available. There has been a large effort in the past few years to make thin film diamond commercially available in thin film form using a technique called chemical vapor deposition (CVD). This process uses a combination of heat and a plasma to extract the carbon from methane gas and then deposit it onto a suitable seed crystal. By varying the processing parameters, electrically insulating and thermally conducting thin films of diamond result. These films are then polished (typically ~300 μm thick) and laser-cut to size.

The process of getting a metal to stick to diamond without it forming a metal carbide interlayer is difficult and not relevant to this research so the films used in this study were supplied with a Ti (1000 Å) / Au (1000 Å) metallization. It was decided that for compatibility with wire-bonding, more gold was necessary and an additional 5000 Å of gold was deposited. In order to make the heat sink compatible with flip-chip bonding, a specific line-and-space pattern was to be transferred into the metallization. For low-frequency applications, these lines could be arbitrary dimensions, but to efficiently pass microwave energy, the lines needed to have a wave impedance near 50 Ω. The style of line chosen for this application was a coplanar waveguide (CPW).

This type of transmission line consists of three conductive stripes, all on the top surface of a dielectric. The center conductor is surrounded by gaps of equal width, which are then surrounded by 'semi-infinite' ground planes. A CPW structure is shown in Figure 1. Using the dielectric constant of the substrate and a center conductor width that was compatible with our probing techniques (44 μm), gap dimensions (10 μm) were determined[9] to give the desired 50 Ω transmission lines.
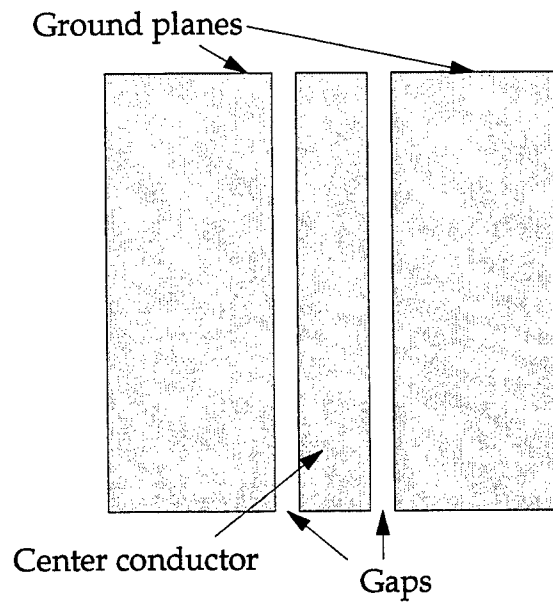
Figure 1: Coplanar Waveguide

With the dimensions determined, photolithography was to be used to transfer the pattern into the metallization. The resulting pattern would be a series of these CPW structures to match a linear array of VCSEL's. There are standard wet chemical etch formulations for both gold and titanium and because of the ease of their use, they were tried first. The side-effect of all wet etching techniques though is undercut. It results from the exposure of the sidewall to the etchant after the etching has begun. In this situation, an approximately 45° angled sidewall profile would be expected.

One important factor that determines the resulting sidewall angle is the adhesion of the photoresist to the underlying film. If the adhesion is not very

good, then there are localized places where the etchant can penetrate further under the resist and this quickly leads to severe undercut. This was observed with the wet chemical etching of the gold film when using a commercially formulated wet etch. For a 6000 Å etch depth, an undercut between .6 μm and 1μm is expected, but for this process, an undercut of 4-6 μm was typical. A diamond heat sink after the gold etch and before the removal of the photoresist is shown in Figure 3.
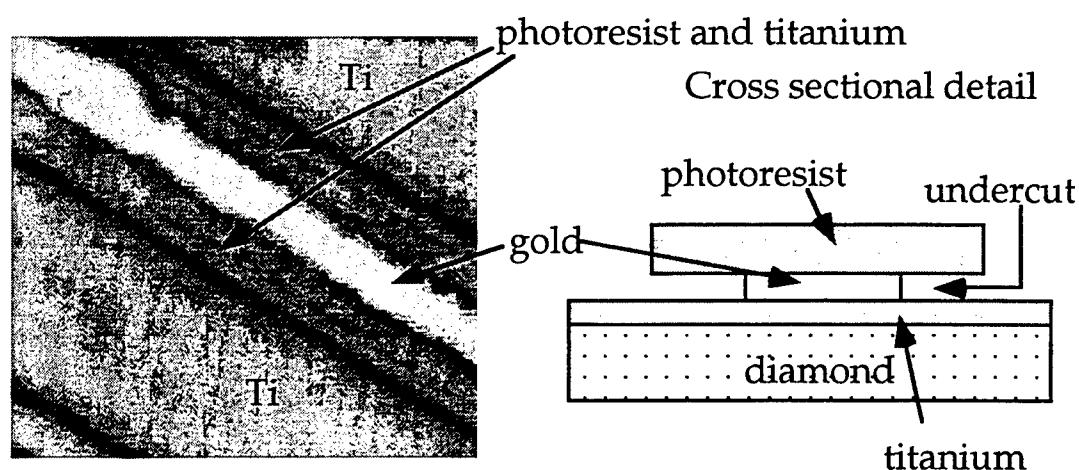
Figure 2: Extreme undercut after the wet etch

The titanium etch (2 ml HF in 100 ml DI H_2O) behaved as expected. The etch is very fast after the initial oxide layer is etched away. After about 8 to 12 seconds in the etch, the exposed titanium surface bubbles and it etched away. The

wafer should then immediately be removed from the HF:DI mixture and rinsed in DI water to minimize undercut.

A dry etch technique for gold was then investigated. There are no known reactive ion etching techniques for gold, but the removal of atoms by the ion bombardment of the surface, sputtering, works well. The technique is called ion milling and consists of an energetic stream of ions (typically argon) directed towards the sample. The ions are neutralized before impacting the sample to eliminate charging effects on insulating samples. The impinging atoms sputter both the photoresist and the gold, so knowledge of the relative sputter rates (gold ~1000 Å/min; photoresist ~250 Å/min) and desired etch depth determines the thickness of photoresist necessary to protect the CPW lines during the etch. To ion mill the 6000 Å of gold so that no isolated traces were visible took 6-3/4 to 7-1/4 minutes, and the endpoint for this etch was easily monitored because of the different colors of gold and titanium. The photoresist mask was removed in an oxygen plasma (~2000 Å/min at 500 V, 30 sccm O_2, 30 W/cm^2) and then the previously mentioned titanium etchant was used to isolate the lines.

Bonding

The bonding of temperature-sensitive devices to heat sinks is very common practice in both the silicon and GaAs industries. Simply soldering the chip to the heat sink using a low melting point solder alloy is the easiest and the most common technique[10,11], but this method leaves the active region, i.e. the top surface of the chip, far away from the heat sink and separated by the low thermal conductivity semiconductor material. Thinning the wafer is an option, but thicknesses below 125 μm are difficult to obtain in a production environment[12]. Flip chip bonding, the technique used in this work, inverts the wafer and bonds the active region to the heat sink. Thus, with flip chip bonding, the distance between the active region and the heat sink is only the thickness of the metallizations, 2 μm in this work. This procedure requires that the heat sink be patterned with conductors for the proper electrical connections to the circuit and a solderable alloy that is compatible with the metallizations on both the heat sink and on the die. Industrially, the alignment tolerances necessary to flip chip bond wafers onto a carrier are compatible with high throughput production, but for this application, tight tolerances on placement and planarity make the process more difficult.

The bonding pads on the die are the tops of the active region of the devices. Thus, care must be taken when choosing the amount of solder to use as well as the pressure that is applied during the bonding sequence. Indium was used as the solder for this project because it has a low melting point (156° C), readily alloys with gold (the metallization on both the heat sink and the laser), and could easily be evaporated to the desired thickness (2 μm). Because a pressure of 4500 psi (3 kg/mm^2) has been found to be safe for bonding GaAs wafers[13], the amount of contact area was determined and it was found that 350 g was the maximum force that could be applied during the bond sequence.

The flip chip aligner/bonder[14] used to mount the wafer onto the diamond offers independent control of the upper and lower chuck temperature setpoints as well as control of the applied pressure during the bond. Upward- and downward-looking cameras were used to align the wafers in X and Y and also in pitch and roll. A user-programmed bonding sequence is then initiated in which the samples are pre-heated, brought into contact, and then held at the specified pressure while the upper and lower chucks are ramped to the reflow temperature (250° C). After reflow, the upper vacuum is released, the lower stage retracts, and the bonded sample is cooled. The packaged bar is shown in Figure 4.
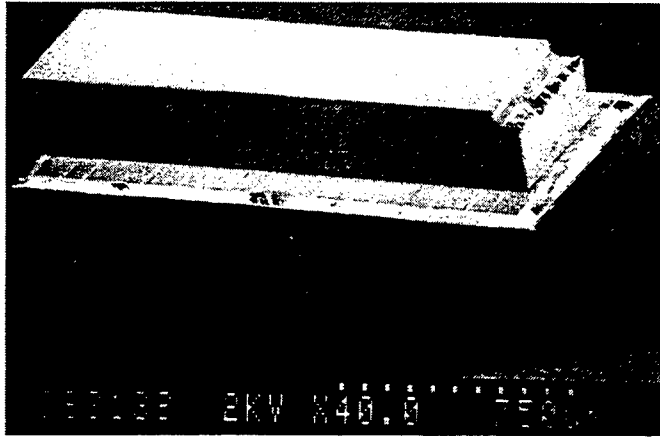
Figure 3: Bonded linear VCSEL array

Microwave test station

To measure the microwave characteristics of these packaged devices, a test station was assembled. The laser was to be modulated using the signal from a vector network analyzer (VNA). The DC component of the signal was supplied by a CW current source attached to a bias tee on port 1 of the VNA. The laser emission was to be collimated using a 0.23 pitch graded index (0.23P GRIN) rod and then focused into a single mode fiber (SMF) pigtailed to a 0.25P GRIN rod. The SMF was attached to a high speed photodetector and the microwave modulation signal was returned to port 2 of the VNA. The setup is shown in Figure 5.

8-12
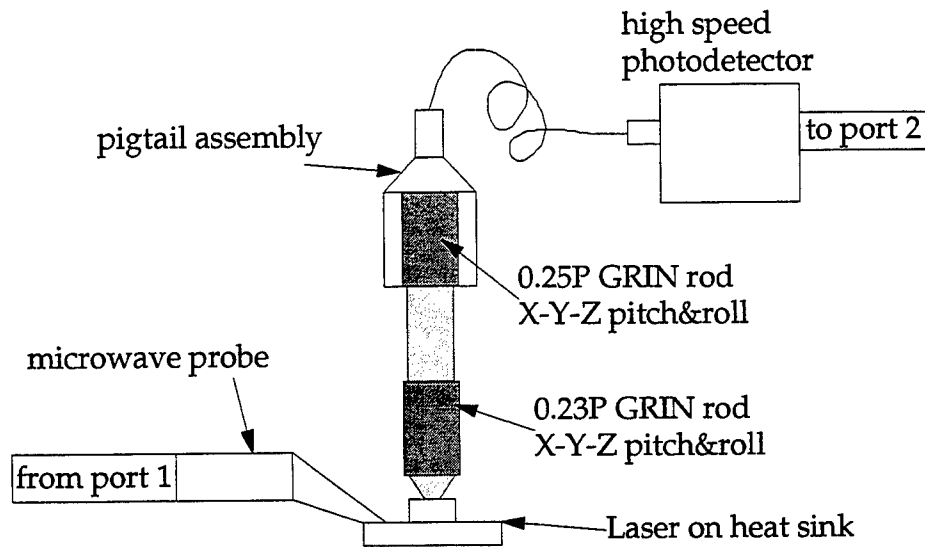
Figure 4: Beam focusing setup

Launching into the SMF is necessary because the coupling onto the 25μm

diameter detector is optimized for a 8μm core SMF. The alignment sequence

starts with simple X-Y-Z positioning of the 0.23P GRIN rod using an IR card to

view the focused beam profile. The collection of light into the SMF was made

easier by first aligning the mounting hardware with a multi-mode fiber pigtailed

to a 0.25P GRIN rod. This assembly was removed and replaced with the single

mode assembly. This always resulted in a relatively large signal coupled into the

SMF with only small variations in pitch and roll and X-Y-Z needed to optimize

the coupling.

Device results

The results from the microwave measurements were very encouraging. The yield for the bonding was near perfect. All probed devices were soldered to the heat sink and all but one lased. Laser spectra and PI curves could be obtained respectively from a fiber-coupled spectrum analyzer and photodetector. The maximum fiber-coupled power for the 10 and 15 μm diameter devices tested was near 250 μW with more typical fiber-coupled powers of 125 to 150 μW. A typical PI curve is shown in Figure 6.
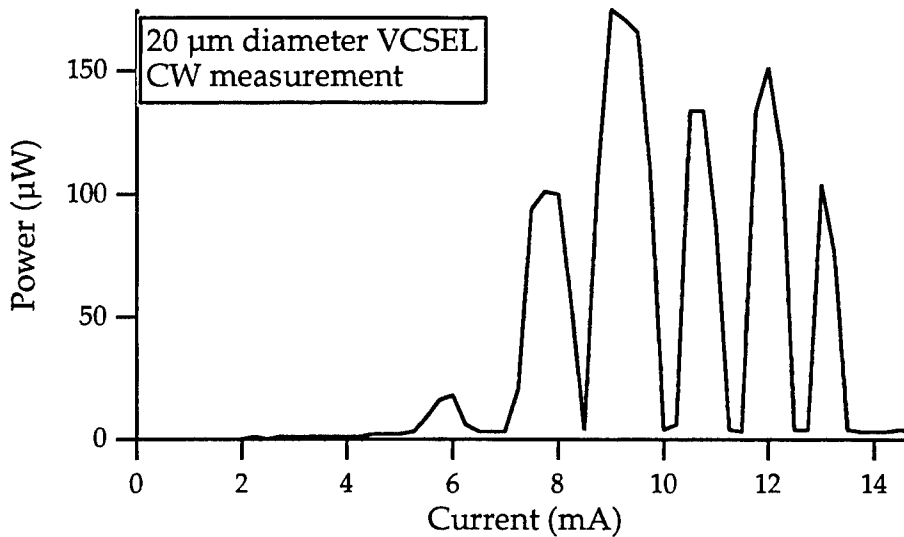
Figure 5: Bonded VCSEL PI curve

An interesting effect is noticed immediately. There are ranges of current in which the lasing mode is extinguished. It was determined that this effect is due to

the external cavity that is formed between the bottom mirror and the polished back surface of the wafer. The reflection couples 30% of the lasing mode back into the cavity, which, when out of phase with the laser mode, will quench the laser action. This effect repeats many times during a PI curve because the wavelength red-shifts with the increased heating at higher bias currents. This small change in wavelengths is enough to shift between the cavity modes of the substrate. Using the refractive index of GaAs at a wavelength of 1 µm and the measured spacing of the maximums in the laser emission, a cavity length of 300 µm was calculated, which is very close to the thickness of the substrate (~0.012").
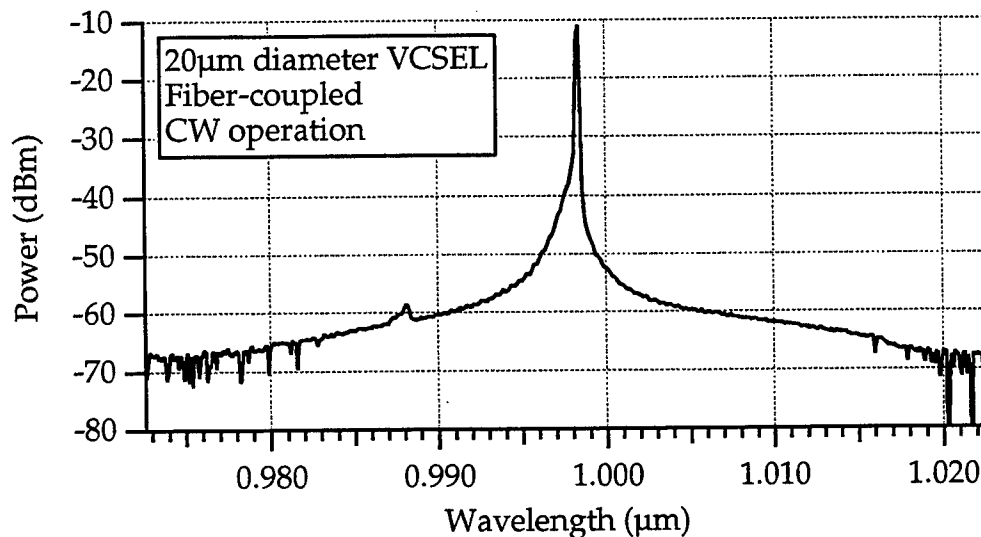
Figure 6: VCSEL Output spectrum

At all power levels, single mode behavior at a wavelength near 1.00 μm was observed for both the 10 μm and 15 μm diameter devices. A spectrum is shown in Figure 7. The power level was seen to drop from near 200 μW at a peak in output power to typically single digit nW at one of the nulls, while the spectrum remained single mode.

The microwave response of a semiconductor laser is very well known and has the following functional form

$$R(\omega) = \frac{\omega_0^2}{(\omega^2 - \omega_0^2) - j\omega\gamma}.$$

Unfortunately the measured response did not follow this form, but contained other resonances. It was learned that the problem was caused by a resonant cavity being set up between the emitting surface of the laser and the bottom of the GRIN rod. The GRIN rod was anti-reflection coated for 830 nm and not 1 μm so its reflectivity was unknown. Unfortunately for this measurement, one of the resonances of that cavity corresponded to where the laser resonance was expected. It was found that by increasing the distance between the laser and the GRIN rod, the resonance would go away, but at a great expense in coupling efficiency. It was then necessary to use a microwave amplifier (with a gain of 18 dBm and a noise figure of 8 dBm) to boost the signal level. After removing that resonant cavity,

8-16

the measured transmission characteristics matched the expected response. Examples of the microwave response with and without the resonance are shown in Figures 8a & 8b.
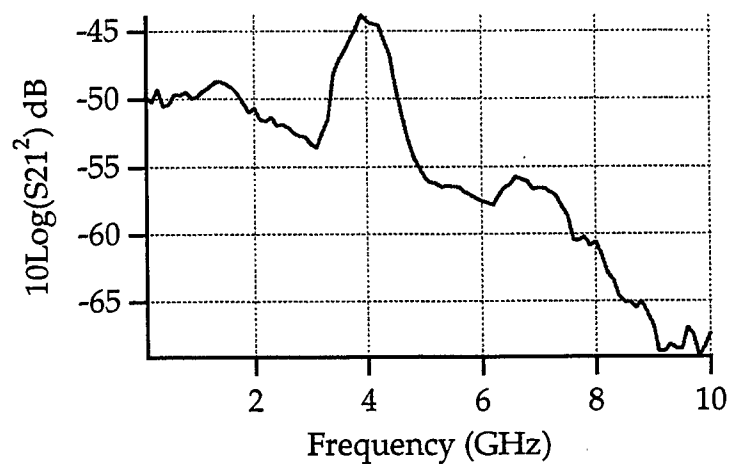
Figure 7a: Microwave response with external cavity resonance
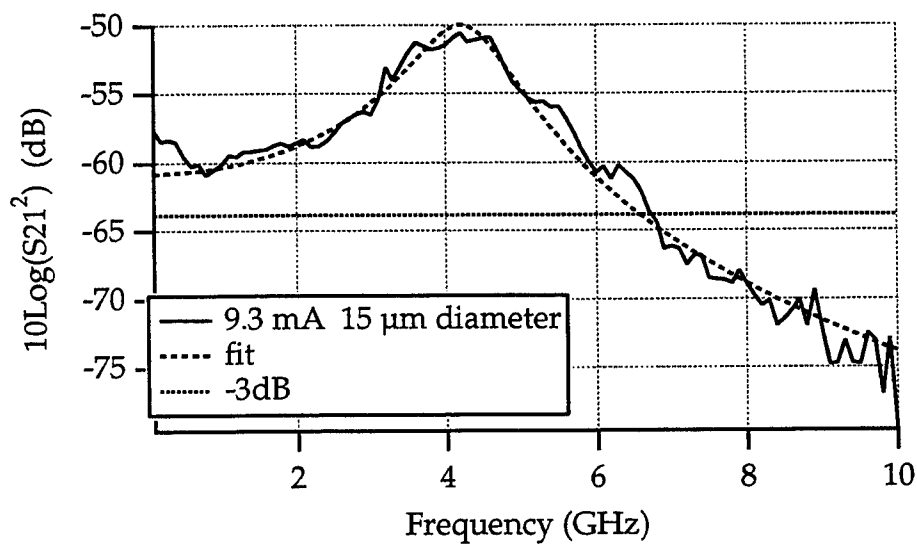
Figure 8b: Microwave response without external cavity resonance

The 15 μm diameter devices had higher resonances and bandwidths than the 10 μm diameter devices. The 15 μm diameter devices typically had a resonance near 4 GHz and a -3dB bandwidth over 6 GHz with the best device (shown in Figure 8) having a resonance at 4.2 GHz and a -3dB bandwidth of 6.6 GHz at 9.3 mA.

Conclusions

Flip-chip bonding was shown to be a reliable technique for attaching these devices to diamond heat sinks. All tested devices were contacted and the package survived handling without failure. The microwave properties of the devices show a packaged bandwidth of over 6 GHz. With minor lithographic changes to the heat sink design used in this report, the packaged devices can be used as a replacement component for the LED's used in the previous Rome Laboratory VOI project.

REFERENCES:

[1] G. Shtengel, H. Temkin, P. Brusenbach, T. Uchida, M. Kim, C. Parsons, W.E. Quinn, and S.E. Swirhun, "High-Speed Vertical-Cavity Surface Emitting Laser", *IEEE Phot. Tech. Lett.*, v. 5, no. 12, pp. 1359-1362, 1993.

[2] P.L. Gourley, T.M. Brennan, B.E. Hammons, S.W. Corzine, R.S. Geels, R.H. Yan, J.W. Scott, and L.A. Coldren, "High-efficiency TEM_{00} continuous-wave (Al,Ga)As epitaxial surface-emitting lasers and effect of half-wave periodic gain", *Appl. Phys. Lett.*, v. 54, n. 13, pp. 1209-1211, 1989.

[3] K. Tai, J.D. Wynn, Y.H. Wang, and A.Y. Cho, "Self-aligned fibre pigtailed surface emitting lasers on Si submounts", *Electron. Lett.*, v. 27, n. 22, pp. 2030-2032, 1991.

[4] D. Vakhshoori, J.D. Wynn, G.J. Zydzik, R.E. leibenguth, M.T. Asom, K. Kojima, and R.A. Morgan, "Top-surface emitting lasers with 1.9 V threshold voltage and the effect of spatial hole burning on their transvers mode operation and efficiencies", *Appl. Phys. Lett.*, v. 62, n. 13, pp. 1448-1450, 1993.

[5] J. Weber, K. Malloy, and S. Wang, "Effects of layer thickness variations on vertical-cavity surface-emitting DBR semiconductor lasers", *IEEE Phot. Tech. Lett.*, v. 2, no. 3, pp. 162-164, 1990.

[6] R. Baets, P. Demeester, and P.E. Lagasse, "High-reflectivity GaAs/AlGaAs mirrors: sensitivity analysis with respect to epitaxial growth parameters", *J. Appl. Phys.*, v. 62, n. 2, pp. 723- 726, 1987.

[7] H.F. Bare, F. Haas, D.A. Honey, D. Mikolis, H.G. Craighead, G. Pugh and R. Soave, "A simple surface-emitting LED array useful for developing free-space optical interconnects", *IEEE Phot. Tech. Lett.*, v. 5, no. 2, pp. 172-175, 1993.

[8] Information about CVD diamond is typical data from Harris Diamond Corporation, 100 Stierli Court Suite 106, Mount Arlington, NJ 07856.

9 J. Braunstein, private communication, 1995.

10 G.S. Matijsevic, C.Y. Wang, and C.C. Lee, "Void free bonding of large silicon dice using gold-tin alloys, " *IEEE Trans. Comp., Hybrids, and Manuf. Technol.,* v. 13, no. 4, pp. 1128-1134, 1990.

11 C.C. Lee, G.S. Matijasevic, "Highly reliable die-attachment on polished GaAs surfaces using gold-tin eutectic alloy," *IEEE Trans. Comp., Hybrids, and Manuf. Technol.,* vol. 12, pp. 406-409, 1989.

12 L.F. Lester, private communication, 1994.

13 L.F. Eastman, private communication, 1995.

14 The Model M-8A Flip Chip Aligner/Bonder is made by Research Devices Piscataway, NY 11111

The Parallelization of a Multitarget Tracking Algorithm for a Class of High-Performance Computing (HPC) Architectures

Robert Popp
Ph.D Candidate
Department of Computer Science and Engineering
rpopp@sol.uconn.edu

University of Connecticut
UBox 155
Storrs, CT 06269-3155

October 5, 1995

The Parallelization of a Multitarget Tracking Algorithm for a Class of High-Performance Computing (HPC) Architectures

Robert Popp
Ph.D Candidate
Department of Computer Science and Engineering
University of Connecticut

Abstract

In recent years, it has been shown to be both viable and economically feasible to use large-scale parallel and distributed high-performance computing (HPC) architectures when solving compute-bound type problems. Within the aerospace community, with the availability of such powerful and affordable parallel processing systems, presently there has been increased interest in the parallelization of "traditionally" computationally-intensive algorithms utilized for multitarget tracking. Multitarget tracking algorithms based on an Interacting Multiple Model (IMM) estimator have been shown to be very effective when applied to tracking maneuvering targets. However, the IMM estimator imposes a computational burden in terms of both space and time complexity, since more than one filter module is used to calculate state estimates, covariances, and likelihood functions.

Hence, in an effort to improve its performance computationally, the primary focus of this report is to present the results of our investigation on the parallelization of an IMM-based multitarget tracking algorithm for a distributed memory, message passing based class of high-performance computers. In particular, using a measurement database based on two FAA air traffic control radars, courtesy of Rome Laboratory, we utilize an Intel Paragon model XP/E HPC as our system platform to develop and evaluate the parallel tracker. We show that good computational performance (in terms of speedup and efficiency) results when utilizing a parallelization technique modeling the supervisor/worker approach and distributing the track data from the multitarget problem randomly and uniformly via the *modulo* scheme. However, since the number of association costs computed, not the size of the local track list, is directly correlated to the nodal computation time, intelligent load balancing schemes are necessary because, in an increasing number of processors, the nodal computation time becomes imbalanced resulting in degraded performance for the multitarget tracker.

1 Introduction

1.1 Motivation

In recent years, it has been shown to be both viable and economically feasible to use large-scale parallel and distributed high-performance computing (HPC) architectures when solving compute-bound type problems. As parallel processors become increasingly used in computing applications, it will become essential to have algorithms available to use them efficiently. Within the aerospace community, with the availability of such powerful and affordable parallel processing systems, presently there has been increased interest in the parallelization of "traditionally" computationally-intensive algorithms utilized for multitarget tracking. The objective of a surveillance and tracking algorithm is to detect an unknown number of targets, in the presence of spurious observations and occasional missed detections, and to estimate their states using sensor measurements of (possibly) unknown origin, contaminated by noise and clutter. Typically, multitarget tracking algorithms demand large computational resources for numerous reasons, including the requirement to identify and track potentially hundreds of targets in real-time. However, as is well documented in the literature [11, 12], these algorithms based on conventional uniprocessor systems generally cannot satisfy such real-time requirements, and therefore adapting such algorithms to HPC architectures poses to be a promising alternative.

The *tracking* performance of a tracking algorithm is mainly governed by the accuracy of the state estimator used. The traditional and most widely used state estimator is the Kalman filter. It provides optimal tracking performance if [4]: (i) the state of the target evolves according to a known linear dynamic model driven by known input, i.e.,

$$x(t_{m_k}) \triangleq F(\delta)\, x(t_{m_{k-1}}) + G(\delta)v \tag{1}$$

where $\delta = t_{m_k} - t_{m_{k-1}}$ is the time interval, $F(\cdot)$ is the state space transition matrix, $G(\cdot)$ is the disturbance matrix, and v is zero-mean, white Gaussian *process* noise with (known) covariance $Q(\cdot)$ (based on a design parameter [15]); and (ii) the measurements are linear functions of the target state corrupted by measurement noise, i.e.,

$$z(m_k) \triangleq H x(t_{m_k}) + w(m_k) \tag{2}$$

where $m_k = 1 \ldots, M(k)$ denotes the m_k^{th} measurement from scan k, $H = [I\ 0\ 0]$ is the measurement matrix, and $w(\cdot)$ is zero-mean, white Gaussian *measurement* noise with (known) covariance $R(\cdot)$. The covariance matrix associated with the target state estimate $\hat{x}(t_{m_k})$ is denoted as $P(t_{m_k})$. However, because of the single motion model assumption, when its design parameters are conservatively modified to track maneuvering targets via increased process noise covariance, the Kalman filter has been found to provide marginal tracking performance [4, 9].

The Interacting Multiple Model (IMM) algorithm [4, 7] has proven to be very effective as a state estimator when tracking maneuvering targets. Inherent to the problem of estimating the states of maneuvering targets

is uncertain target dynamics; the motion of such targets cannot be modeled well via a single set of state equations. To achieve superior tracking performance for maneuvering targets, it is essential that the estimator allow for the motion of such targets to be described by different state equations in different time intervals [9]. The IMM estimator, consisting of a finite number of filters each modeling different target motions, contains an explicit provision for the target motion to probabilistically "switch" from one motion model to another[1]. One common approach to implementing the IMM estimator is to use a bank of Kalman filters, where each filter is driven by the same set of measurements, but differ in the values for process noise covariance to allow for target maneuvers.

Extensive evidence [5, 8, 9] strongly suggests that, when tracking maneuvering targets, an IMM estimator provides better performance, in terms of tracking accuracy, than a well-tuned Kalman filter. However, because of the additional filters, it also has a higher computational load. For any IMM-based tracking algorithm, there is a requirement to have a reasonable number of filter modules to cover the possible target maneuvers. However, to be practical in a real-time environment, the computation time needs to be considered, since increasing the number of filter modules increases the computational burden of the IMM estimator considerably. It is also theoretically shown in [10] that the use of too many modules is as bad as the use of too few in terms of tracking accuracy[2]. Thus, it is desirable to use a moderate number of filter modules in the IMM estimator to allow for both good tracking and computational performance. However, to date, there has been a lack of efficient parallelizations of IMM-based tracking algorithms reported in the literature using a moderate number of filter modules and filling this gap is one of the primary focuses of the present work.

1.2 Related Research

In this subsection, we briefly review some related work in this area. In terms of parallel multitarget tracking algorithms, Pattipati *et al.* [12] describe efficient mappings of model-based multitarget tracking algorithms (e.g., multiple hypothesis tracking (MHT) type algorithms) onto MIMD multiprocessors. They clearly show that the task granularity and processor architecture are major determinants of speedup of a multitarget tracking algorithm. Atherton *et al.* [1] describe a parallel algorithm to track multiple targets using a branch- or track-splitting algorithm, where the multiple tracking filters are mapped onto a transputer, a message-passing architecture. However, track-splitting type algorithms are suspect in real-time environments, since many filters may have to be run when branching or splitting occurs, thus making them computationally demanding, especially in cluttered scenarios.

---

[1]By necessity, we assume that the target obeys one of a finite number of models.

[2]Unsatisfactory *tracking* performance may result in an IMM estimator with a large number of filter modules since many of the modules will differ significantly from the system mode in effect at a particular time yielding excessive "competition" from the "unnecessary" filter modules. For "higher dimensional" systems, where a large number of filter modules may be necessary, an IMM estimator with a *variable-structure*, where only a relatively small *fixed* number of filter modules are active at any particular time, proves to be an excellent alternative [10].

Atherton *et al.* [2] develop a "fine-grained" parallelization of an IMM-based tracking algorithm on a 4-processor transputer architecture. Essentially the multiple filter modules of the IMM estimator are run in parallel, while the other components of the estimator (i.e., interaction, update, and combination steps) are run sequentially. Using an IMM estimator with 9 and 13 filter modules, they obtained speedups (efficiencies) [3] of 2.34 (59%) and 2.73 (68%), respectively, while not sacrificing tracking accuracy. Using a *time-slipping* heuristic where some tracking accuracy was traded off for increased computation performance, speedups (efficiencies) of 3.11 (78%) and 3.21 (80%), were achieved. Averbuch *et al.* [3] develop a fine-grained parallelization of an IMM-based tracking algorithm on a (4-processor) MIMD multiprocessor. For an IMM estimator designed with 12 filter modules, they were able to obtain a speedup (efficiency) of 3 (75%). In both of these works, the parallelization of the tracking algorithm is static (fixed) *within* the IMM, and, as we demonstrated in [13], is more appropriate for single target tracking.

1.3 Scope and Organization of the Paper

The primary focus of this report is to present the results of our investigation on parallelizing an IMM-based multitarget tracking algorithm for a distributed memory, message passing based class of high-performance computers. In particular, using a measurement database based on two FAA air traffic control radars, courtesy of Rome Laboratory, we utilize an Intel Paragon model XP/E HPC as our system platform to develop and evaluate the parallel tracker. The parallelization that we primarily discuss in this report can be characterized as utilizing the supervisor/worker model, and, as such, is inherently centralized. Hence, critical to the efficient computational performance of such a parallelization is minimizing the serial portion of the supervisor node's work and evenly distributing the workload across the worker node set (i.e., maintaining load balance in terms of nodal computation time). In the current parallelization, a random and uniform *modulo* track data distribution scheme was utilized, with parallel efficiencies of $75\% \pm 5\%$ when using between $4-25$ processors. However, because the number of association cost computations (where a likelihood function is evaluated) that a node executes drives its computation time and not the number of tracks that it must process, a slight degradation in parallel efficiency resulted in our present parallelization. Future works are addressing this deficiency via intelligent load balancing schemes and alternative distributed algorithms.

The remainder of this paper is organized as follows. Section 2 provides an outline of the multitarget tracking algorithm used in this work, while Section 3 discusses the parallel design of our multitarget tracker. Section 4 presents analytical analysis and performance results of our parallel multitarget tracking algorithm based on the FAA measurement database. Section 5 concludes the paper by summarizing our findings.

---

[3]The standard definitions of speedup and efficiency are used, i.e., *speedup* $\triangleq \frac{\tau_1}{\tau_p}$, where τ_1 (τ_p) denotes the sequential (parallel) execution time utilizing 1 (p) processor(s), and *efficiency* $E_p \triangleq \frac{speedup}{p}$, respectively.

```
t
│
↓
┌─────────────────┐                          Work Distribution
│ Obtain scan     │                    ┆     ─────────────────
│ measurements    │                    ┆           .7%
└─────────────────┘                    ┆
        ↓                              ┆
┌─────────────────┐                    ┆
│ Compute measurement                  ┆
│ to track association costs           ┆         94.7%
│ with IMM estimator                   ┆
└─────────────────┘                    ┆
        ↓                              ┆
┌─────────────────┐                    ┆
│ Global data association of           ┆
│ measurement and track data           ┆          .1%
└─────────────────┘                    ┆
        ↓                              ┆
┌─────────────────┐                    ┆
│ Track formation and/or               ┆
│ track extension updates              ┆         4.5%
└─────────────────┘                    ┆
        │
      t+1
```
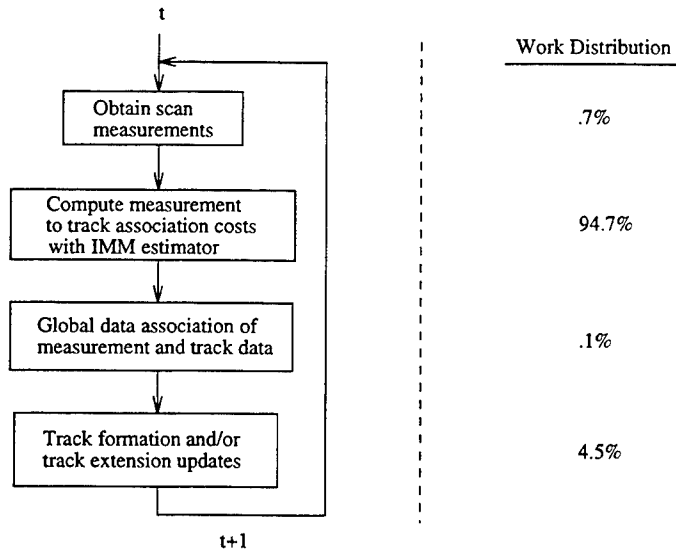
Figure 1. Block diagram and work distribution of the multitarget tracking algorithm MATSurv II.

2 Multitarget Tracking Algorithm Outline

Developed in [14, 15] is an experimental multitarget Multisensor Air Traffic Surveillance (MATSurv) algorithm which utilizes a decoupled IMM estimator in conjunction with a 2-dimensional (2-D) sliding window assignment algorithm[4]. Besides the ancillary tasks of *obtaining scan measurements* and *track formation/extension*, the major tasks of a multitarget tracking algorithm can be broadly divided into two interrelated categories: *data association* and *state estimation*. The data association problem in MATSurv involves an optimal assignment of measurements from the latest scan to the most likely tracks from the previous scans using a global cost function (based on a *maximum likelihood* criterion as opposed to a *maximum a posteriori* criterion approach combined with an enumerative search employed in MHT techniques). The state estimation problem in MATSurv provides: (i) a measure of how likely it is that a particular measurement from a given scan originated from a particular target, i.e., provides a *measurement-target* association cost that we utilize in the data association problem, and (ii) an estimate of the target state. Figure 1 shows a control-flow block diagram of our multitarget tracking algorithm in terms of the four tasks of obtaining scan measurements, state estimation, data association, and track formation/extension and their respective workload distributions (more discussion about this later). Appendix A briefly details how the state estimation problem was addressed in MATSurv II and we refer interested readers to [15] for a more thorough description.

---

[4] In [14], MATSurv I relied on Kalman filters as state estimators. MATSurv II, presented in [15], used IMM state estimators with Kalman filter modules; moreover, decoupling the filters in the horizontal and vertical planes enabled the IMM estimator to handle a target maneuver in one plane, while correctly recognizing the lack of any maneuver in the other plane, if such were the case. The present work parallelizes MATSurv II.

Here we briefly describe how the maximum likelihood data association problem was formulated as a sliding window 2-D assignment problem [14, 15]. In the following section, we motivate our work in parallelizing the IMM-based multitarget tracking algorithm by describing the computational significance of computing such measurement-target association costs in setting up the 2-D assignment problem.

Association is the decision process of *linking* measurements (from successive scans) of a common origin (i.e., a target or false alarms) such that each measurement is associated with only one origin. In the present formulation, $M(k)$ measurements from the latest scan k are to be assigned to the $N(k-1)$ most likely existing tracks from the previous scans using a global cost minimization [5]. Let $n = 0, \ldots, N(k-1)$ denote a particular track from the "list" of existing tracks (including a *dummy* track $n = 0$), and $m_k = 0, \ldots, M(k)$ denote a particular measurement from the latest scan "list" (including a *dummy* measurement $m_k = 0$). Define the binary "assignment" variable

$$\omega(n, m_k) = \begin{cases} 1 & \text{if detection } m_k \text{ is assigned to track } n \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Note that $\omega(0, m_k) = 1$ implies that detection m_k is not assigned to any of the $N(k-1)$ previously established tracks, and $\omega(n, 0) = 1$ implies that track n has missed a detection at scan k. Since the measurements $z(m_k)$, $m_k = 1, \ldots, M(k)$ are independent of each other, maximizing the likelihood function, consisting of the joint pdf-probability [4] of measurements given their origins and the corresponding detection events, over the set of feasible assignments[5] can be cast into the following assignment problem:

$$\text{minimize} \quad \sum_{n=0}^{N(k-1)} \sum_{m_k=0}^{M(k)} \omega(n, m_k) \, c(n, m_k) \tag{4}$$

$$\text{subject to:} \quad \sum_{m_k=0}^{M(k)} \omega(n, m_k) = 1 \quad n = 1, \ldots, N(k-1) \tag{5}$$

$$\sum_{n=0}^{N(k-1)} \omega(n, m_k) = 1 \quad m_k = 1, \ldots, M(k) \tag{6}$$

where $c(n, m_k)$, the "cost" of assigning detection m_k to track n, is given by

$$c(n, m_k) \stackrel{\triangle}{=} -\log\left(\frac{\Lambda(n, m_k)}{\Lambda(0, m_k)}\right) \tag{7}$$

where the numerator, obtained from the IMM state estimator, is the likelihood that the m_k^{th} measurement at scan k originated from the nth track as defined by Eq. (12) in Appendix A, and the denominator is the likelihood that the m_k^{th} measurement corresponds to none of the existing tracks and is assumed uniformly probable over the sensor surveillance volume [14]. The global minimization in Eq. (4) subject to the constraints in Eqs. (5)–(6) is the generalized 2-D assignment problem and is solved in MATSurv via a polynomial-time modified Auction algorithm [6].

---

[5] An assignment is *feasible* if: (i) each track (excluding the dummy track) is assigned at most one measurement, (ii) each measurement (excluding the dummy measurement) is assigned to at most one track (including the dummy track), and (iii) the dummy measurement is not assigned to the dummy track.

Intel Paragon HPC Architecture	
System Interconnect	
Compute partition size	25 nodes
Interconnection topology	2-D Mesh
Communications paradigm	Message passing
Message latency (hardware)	40 ns/hop across mesh
Node-to-node bandwidth	200 MB/sec full duplex
Node Specifications	
Processors per node	2 application processors; 1 messaging coprocessor
Processor technology	50 MHz (75 MFLOPS) i860 XP RISC processor
On-chip cache	16 KB instruction; 16 KB data
Second-level cache	256 KB
RAM	64 MB/node
Memory bus bandwidth	400 MB/sec

Table 1. Intel Paragon specifications.

3 Parallel Multitarget Tracker Design

As shown in Figure 1, the vast majority of the processing time spent per scan in the sequential implementation of our multitarget tracker involves computing the set of measurement-target (IMM-based) association costs in setting up the data association problem: this constitutes 94.7% of the workload. Besides computing maximum speed and elliptical gates [5] for the target, the association cost is dominated computationally by the negative log-likelihood function evaluation given by Eq. (7)[6]. Moreover, increasing the number of filter modules in the IMM estimator only increases the dominance of the likelihood function evaluation. Clearly, because computing such association costs is the major computational bottleneck, this is where we focus on using efficient parallelization techniques in the present work; however, by the very design of our parallel multitarget tracker, updates and/or initializations of tracks are also done in parallel.

3.1 Implementation Environment

Hardware specifications for the Intel Paragon used in this work is provided in Table 1. We should note that even though 25 nodes were available in the compute partition, a subset of these nodes can be requested at runtime for executing the parallel multitarget tracker (for which we take advantage of when benchmarking

---

[6] Elliptical gating is performed only if the measurement falls within the target's maximum speed gate, and the likelihood function is evaluated *only* if the measurement falls within *both* target gates.

the tracker). Moreover, *gang scheduling* of nodes is strictly employed in the compute partition. In a gang scheduled partition, an application has sole access to the set of nodes it executes on (i.e., no time-sharing of nodes across multiple applications is allowed). Furthermore, besides the messaging coprocessor, only one of the two application processors within each node are utilized. Moreover, nodal communication (via the messaging coprocessor) and computation (via the application processor) were not overlapped in the present parallelization. And finally, the Paragon runs a subset of the standard OSF/1 UNIX operating system (OS), Paragon OSF/1, that is fully distributed, runs in parallel across all nodes of the system, and is AT&T System V.3, 4.3BSD, and POSIX compliant.

3.2 Parallel Design Description

The parallelization of the multitarget tracking algorithm that we designed in this work can be characterized as utilizing the supervisor/worker model, that is, the multitarget problem is decomposed across a worker node set on the HPC and the behavior of the worker nodes is controlled via a dedicated supervisor node. Specifically, the data association problem is solved in parallel by assigning to each worker node a subset of the list of tracks, where the set of worker nodes' track lists are mutually exclusive and exhaustive, and the worker nodes independently (and concurrently) compute measurement-target association costs based on their local track lists. Once finished, each worker node asynchronously sends back to the supervisor node a list of 3-tuples consisting of track IDs from their track list, measurement ID, and measurement-target association cost. Once the supervisor node collects all such lists of 3-tuples from the set of worker nodes, it then solves the "global" assignment problem. After resolving the assignment of tracks to measurements, the supervisor node then broadcasts the assignment to all worker nodes, where each worker updates its local tracks and/or initializes new tracks based on the assignment. In Figure 2 we provide a task graph to illustrate the parallelization as just described.
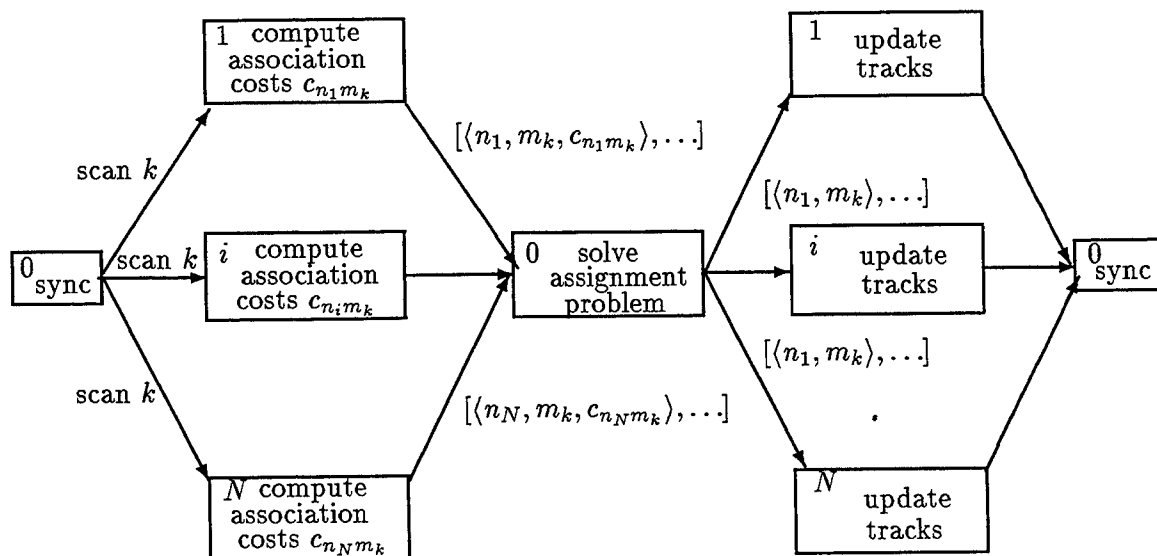

Figure 2. Task graph of parallel multitarget tracking algorithm.

As is evident, a parallelization based on the supervisor/worker model is inherently centralized, and key to its efficient computation performance is minimizing the serial portion of the supervisor node's work and evenly distributing the workload across the worker node set (i.e., maintaining load balance in terms of nodal computation time). In the present parallelization, given 1 node as supervisor, say node 0, and N nodes as workers, say nodes $1, \ldots, N$, we utilize a *modulo* scheme in distributing tracks across the worker node set. That is, for $i = 1 \ldots N$, worker node i has tracks $T_i = \{n : n \mod N + 1 = i\}$. Hence, each worker node is responsible for computing association costs based on its local track list, that is, each worker node computes association cost $c_{n_i m_k}$, where n_i denotes the nth track in T_i and m_k denotes the mth measurement from scan k. The *modulo* scheme provides for a random distribution of tracks across the worker node set in terms of actual target tracks vs. those tracks created based on false alarms. Moreover, a uniform distribution of tracks in terms of track allocation across the worker node set is provided for by this scheme as well; however, a non-uniform distribution of computation time across the worker node set resulted in our parallelization because the number of likelihood function evaluations that a node has to compute drives its computation time, not the number of tracks that it must process. Moreover, the number of likelihood function evaluations that a node computes is a function of the particular tracks local to the node, the sensor from which the measurements were observed, and the number of local active tracks at the node. As we describe later, a slight degradation in parallel efficiency resulted in our present parallelization because our workload distribution was in terms of the number of tracks the node processes, and not the number of likelihood function evaluations that the node must execute. Future works are addressing this deficiency via intelligent load balancing schemes and alternative distributed algorithms.

4   Analysis and Results

In this section we present various performance results and analysis that we obtained based on executing the parallel multitarget tracker on the FAA measurement database. Important aspects of the raw scan measurement database can be found in [13, 14, 15]. For purposes of this paper, what is important to know is that the multitarget scenario captured via the two FAA air traffic control radars is relatively sparse[7]. The data consist of scans, arriving approximately every 10 seconds, from two L-band FAA radars located at Remsen and Dansville, NY. The full database consists of 210 scans (98 from Remsen & 112 from Dansville), while a modified version of the database (for which we report in this paper) consists of 55 scans (26 from Remsen & 29 from Dansville). To illustrate the sparsity of the database, in Figure 3 we plot the normalized distributions of measurement-target associations having a likelihood function evaluated vs. those associations eliminated from a likelihood function evaluation due to the gating schemes. We plot in Figure 4, as a function of the scan number, the number of tracks, number of measurements, the total number of possible measurement-target associations, and the number of measurement-target associations where a likelihood

---

[7] We should note that all results reported will also hold for a dense scenario; however, parallelization of the global assignment algorithm, (which requires more than a small fraction of the total computation) is also needed.

function was evaluated. Note that, even in this relatively sparse environment, there are hundreds of potential associations, per scan, where likelihood functions are evaluated.
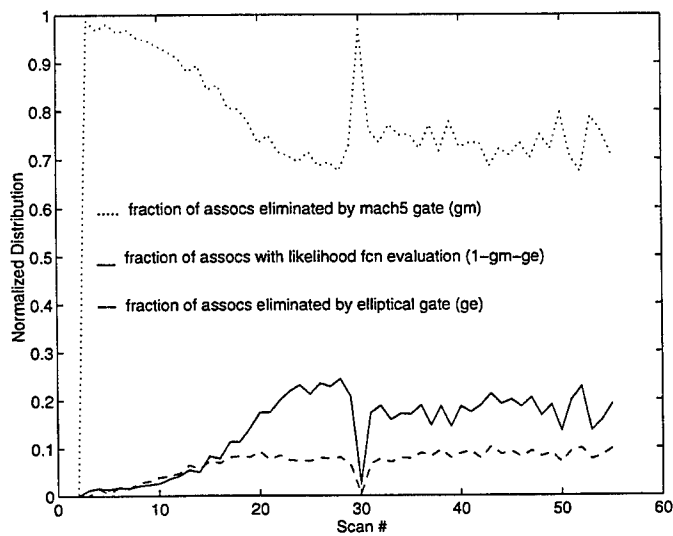
Figure 3. Distribution of measurement-target associations having a likelihood function evaluated vs. those associations eliminated from a likelihood function evaluation due to the gating schemes.
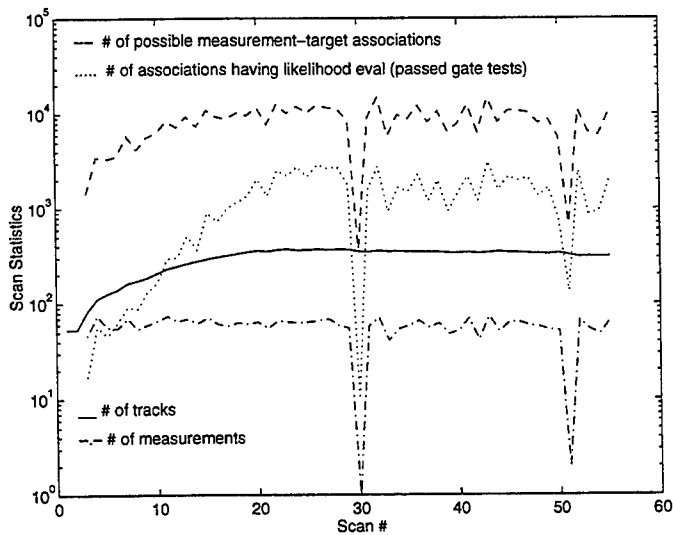
Figure 4. Comparison of various scan statistics.
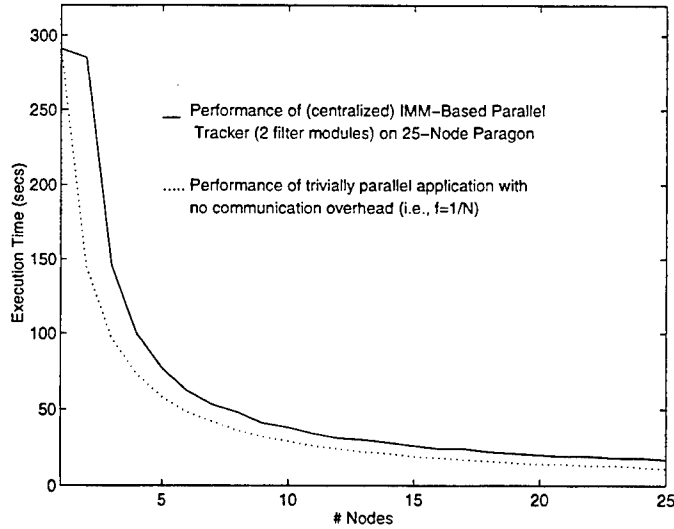
Figure 5. Execution time vs. # of nodes used on 25-Node Paragon.

In Figure 5, we plot, over the number of nodes p utilized on the Paragon, the execution time τ_p of the parallel multitarget tracker and its execution time if it were to obtain linear speedups (e.g., trivially parallel with no communication overhead). Clearly, because $\approx 98\%$ of the workload in the multitarget tracker is parallelized, in addition to the fact that minimal communication between the supervisor node and the worker nodes is performed, the parallelization of the multitarget tracker as proposed in this work (via the supervisor/worker model) has demonstrated that it is fairly efficient.
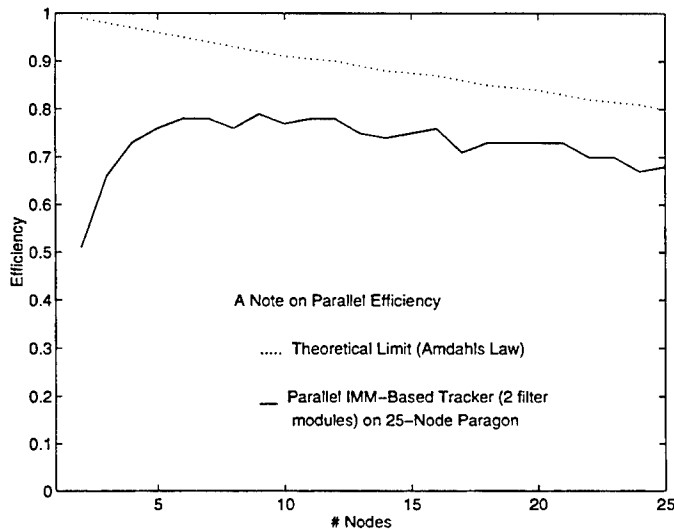
Figure 6. Efficiency vs. # of nodes used on 25-Node Paragon.

However, to determine how well we utilize the multiple processor resources in the proposed parallelization, we plot, in Figure 6, the efficiency E_p of the multitarget tracker. In addition, based on *Amdahl's Law* which

states that parallel efficiency is theoretically bounded by the serial portion of the algorithm, we plot in Figure 6 Amdahl's formulation for efficiency:

$$E_p \overset{\Delta}{=} \frac{1}{\beta p + (1 - \beta)}$$

where β denotes the serial portion of algorithm, and $(1 - \beta)$ denotes the parallel portion (we take $\beta = .01$ in Figure 6). Again, we see fairly stable performance, in terms of utilization of processor resources, for the parallel multitarget tracker; however, E_p drops off somewhat for larger values of p. Certainly parallel efficiency is a function of numerous variables, including the computer architecture, number of processors used, computation-to-communication ratio, problem decomposition and data distribution, fraction of the algorithm parallelizable, and, in particular, the load balance:

$$\mathcal{L} \overset{\Delta}{=} 1 - \frac{X_{max} - X_{min}}{X_{max}}$$

where X is the quantity of interest to measure.
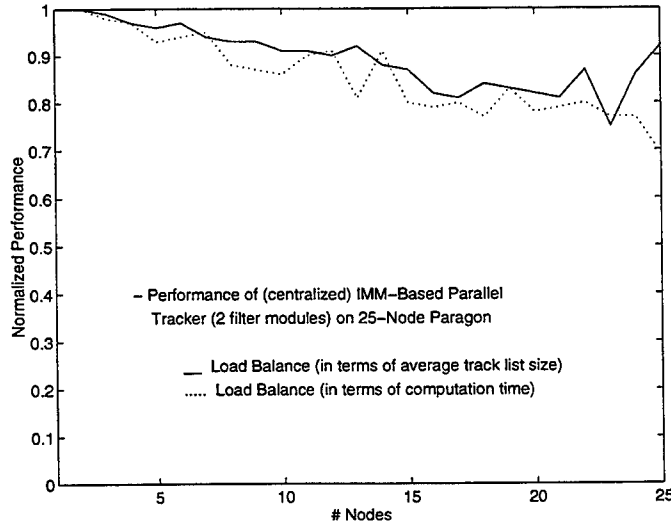
Figure 7. Load balance: nodal computation time vs. average track list size (IMM=2) on 25-Node Paragon.

To determine how well our workload is distributed across the worker node set, in Figures 7 & 8, we plot, over the number of nodes p utilized for the two cases of the IMM having two and five filter modules, the load balance in terms of nodal computation time (i.e., total time spent by the node computing, not communicating) and average nodal track list size. Clearly, even though there apparently is a general tendency for the two functions to behave similarly, there certainly does not seem to be a direct correlation between the two. Hence, our uniform track data distribution scheme via the *modulo* operator may in fact not be very efficient in terms of nodal computation time load balancing. Furthermore, we can see from the plots a degradation in $\mathcal{L}$ as p increases, similarly as was the case for E_p.
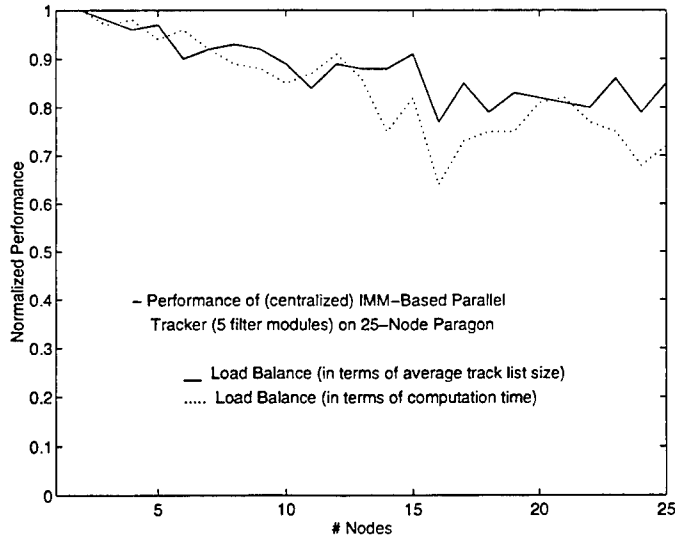
Figure 8. Load balance: nodal computation time vs. average track list size (IMM=5) on 25-Node Paragon.
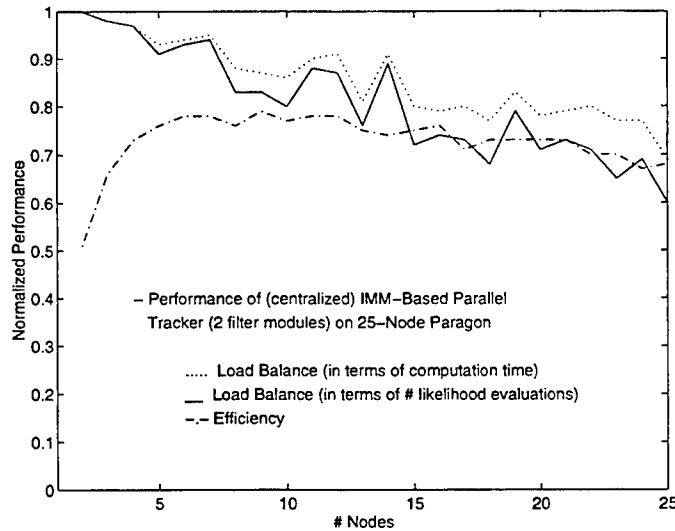
Figure 9. Load balance: nodal computation time vs. number of costs c_{nm_k} computed (IMM=2).

To determine what causes the load balance in terms of nodal computation time to behave as it does, in Figures 9 & 10, we plot, over the number of nodes p utilized for the two cases of the IMM having two and five filter modules, the load balance in terms of nodal computation time and the number of association costs c_{nm_k} computed (where a likelihood function $\Lambda(\cdot)$ based on the IMM estimator was evaluated). Clearly, these two functions are correlated, that is, the computational load balance is driven by the "balance" of the c_{nm_k} distribution across the worker node set. Hence, improving the nodal balance in terms of the c_{nm_k} distribution logically should improve the nodal computational balance, which in turn should improve parallel efficiency (by some factor) since efficiency is a function (in part) of the computational load balance.
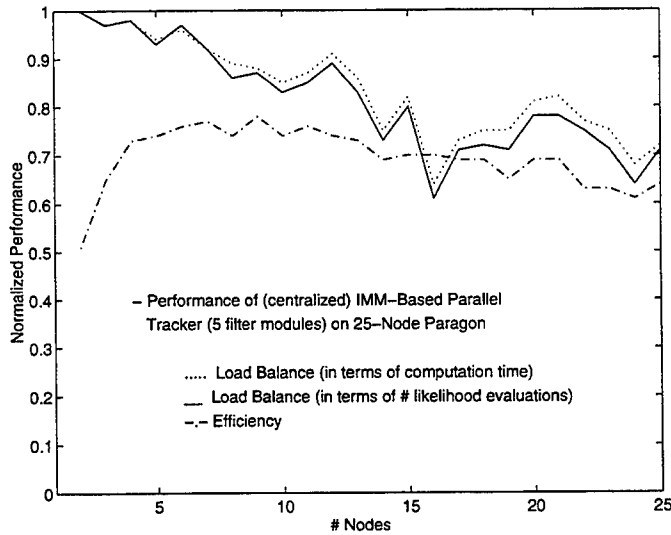
Figure 10. Load balance: nodal computation time vs. number of costs c_{nm_k} computed (IMM=5).

5   Conclusions

In this report, we have presented the results of our investigation and evaluation of parallelizing an IMM-based multitarget tracking algorithm for a distributed memory, message passing based class of high-performance computers. In particular, a measurement database, courtesy of Rome Laboratory, based on two FAA air traffic control radars served as the data for our multitarget scenario and the Intel Paragon model XP/E HPC served as our system platform to develop and evaluate the parallel tracker. We demonstrated that utilizing a parallelization technique modeling the supervisor/worker approach resulted in good computational performance (in terms of speedup and efficiency) for the parallel multitarget tracker. However, because the nodal computation time became imbalanced in an increasing number of processors utilized in solving the multitarget problem, the random and uniform *modulo* track data distribution scheme may not be an optimal technique. Hence, as part of future works, addressing this deficiency via intelligent load balancing schemes and alternative distributed algorithms will be pursued.

A State Estimation
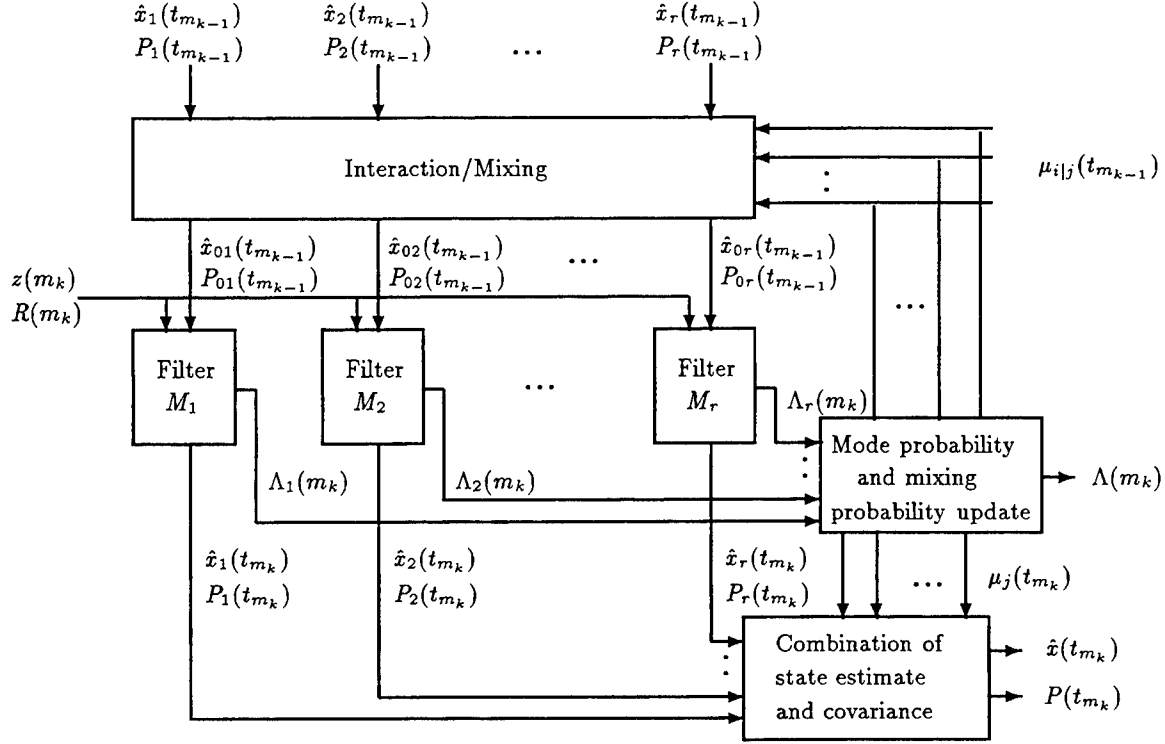
Figure A_2. Generalized IMM algorithm.

The Interacting Multiple Model (IMM) estimator used in the multitarget tracker MATSurv II [15], illustrated in Figure A_2, is a generalized version of the IMM estimation algorithm [4]. As Figure A_2 depicts, the various computations involved in one cycle of the IMM estimator can be divided into: (i) interaction/mixing, (ii) filtering/prediction, (iii) update, and (iv) combination. When using an IMM estimator, the motion of a target for which a corresponding track will develop is assumed to follow one of r possible filter models (or modes) between two successive detections. The version of the IMM estimator used in our work differs from a conventional IMM estimator [4, 5] in that it combines the likelihoods $\Lambda_j(\cdot)$, $j = 1, \ldots, r$ from the individual mode-matched filters to yield a likelihood $\Lambda(\cdot)$ for the overall IMM estimator. Moreover, this particular version of the IMM estimator is specifically designed to track targets in an ECI (Earth centered inertial) reference frame over large distances, with measurements originating from multiple, geographically separated, sensors. Below we omit the intricate details of the generalized IMM algorithm used in the multitarget tracker since it is not the main focus of this paper, and simply present a brief summary of the relevant equations and refer interested readers to [15] for a more descriptive presentation of the material.

The *mixed state estimate* and *covariance* are given by

$$\hat{x}_{0j}\left(t_{m_{k-1}}\right) = \sum_{i=1}^{r} \hat{x}_i\left(t_{m_{k-1}}\right)\mu_{i|j}\left(t_{m_{k-1}}\right) \tag{8}$$

$$P_{0j}\left(t_{m_{k-1}}\right) = \sum_{i=1}^{r} P_i\left(t_{m_{k-1}}\right)\mu_{i|j}(t_{m_{k-1}}) +$$

$$\sum_{i=1}^{r}\left[\hat{x}_i\left(t_{m_{k-1}}\right) - \hat{x}_{0j}\left(t_{m_{k-1}}\right)\right]\left[\hat{x}_i\left(t_{m_{k-1}}\right) - \hat{x}_{0j}\left(t_{m_{k-1}}\right)\right]'\mu_{i|j}(t_{m_{k-1}})$$

(9)

The *mixing probability* is given by

$$\mu_{i|j}(t_{m_{k-1}}) = \frac{\mu_i(t_{m_{k-1}})\pi_{ij}}{\sum_{i=1}^{r}\mu_i(t_{m_{k-1}})\pi_{ij}}$$

(10)

where $\pi_{ij} \triangleq P\left\{u(t_{m_k}) = j | u(t_{m_{k-1}}) = i\right\}$ is the model transition probability (based on a design parameter).

The *mode probability* is given by

$$\mu_j(t_{m_k}) = \frac{\Lambda_j(m_k)\sum_{i=1}^{r}\mu_i(t_{m_{k-1}})\pi_{ij}}{\Lambda(m_k)}$$

(11)

The *combined likelihood of the IMM estimator* is given by

$$\Lambda(m_k) = \sum_{j=1}^{r}\sum_{i=1}^{r}\Lambda_j(m_k)\mu_i(t_{m_{k-1}})\pi_{ij}$$

(12)

and each filter model *likelihood function* is given by

$$\Lambda_j(m_k) = |2\pi S_j(m_k)|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\alpha_j(m_k)\right\}$$

(13)

where the *normalized innovation squared* is

$$\alpha_j(m_k) = \nu_j(m_k)'\left[S_j(m_k)\right]^{-1}\nu_j(m_k)$$

(14)

and the *measurement residual* and *residual covariance* are

$$\nu_j(m_k) = z(m_k) - H\,F_j(\delta_k)\hat{x}_{0j}\left(t_{m_{k-1}}\right)$$

(15)

$$S_j(m_k) = R(m_k) + H\left[F_j(\delta_k)P_{0j}\left(t_{m_{k-1}}\right)F_j(\delta_k)' + G_j(\delta_k)Q_j(\delta_k)G_j(\delta_k)'\right]H'$$

(16)

where $\delta_k = t_{m_k} - t_{m_{k-1}}$.

The *updated state estimate* and *covariance* are given by

$$\hat{x}_j\left(t_{m_k}\right) = F_j(\delta_k)\hat{x}_{0j}\left(t_{m_{k-1}}\right) + W_j(m_k)\,\nu_j(m_k)$$

(17)

$$P_j\left(t_{m_k}\right) = \left[F_j(\delta_k)P_{0j}\left(t_{m_{k-1}}\right)F_j(\delta_k)' + G_j(\delta_k)Q_j(\delta_k)G_j(\delta_k)'\right] - W_j(m_k)\,S_j(m_k)\,W_j(m_k)'$$

(18)

where the filter gain is

$$W_j(m_k) = \left[F_j(\delta_k) P_{0j}\left(t_{m_{k-1}}\right) F_j(\delta_k)' + G_j(\delta_k) Q_j(\delta_k) G_j(\delta_k)' \right] H' \left[S_j(m_k) \right]^{-1} \tag{19}$$

And finally, the *combined state estimate* and *covariance* are given by

$$\hat{x}\left(t_{m_k}\right) = \sum_{j=1}^{r} \hat{x}_j\left(t_{m_k}\right) \mu_j(t_{m_k}) \tag{20}$$

$$P\left(t_{m_k}\right) = \sum_{j=1}^{r} P_j\left(t_{m_k}\right) \mu_j(t_{m_k}) + \tag{21}$$

$$\sum_{j=1}^{r} \left[\hat{x}_j\left(t_{m_k}\right) - \hat{x}\left(t_{m_k}\right) \right] \left[\hat{x}_j\left(t_{m_k}\right) - \hat{x}\left(t_{m_k}\right) \right]' \mu_j(t_{m_k})$$

References

[1] D. Atherton, E. Gul, A. Kountzeris and M. Kharbouch, "Tracking Multiple Targets Using Parallel Processing", *IEE Proc. -Control, Theory and Applications.*, vol. 137, no. 6, pp. 225-234, July 1990.

[2] D. Atherton and H. Lin, "Parallel Implementation of IMM Tracking Algorithm Using Transputers", *IEE Proc. -Radar, Sonar Navig.*, vol. 141, no. 6, pp. 325-332, December 1994.

[3] A. Averbuch, S. Itzikowitz and T. Kapon, "Parallel Implementation of Multiple Model Tracking Algorithms", *IEEE Trans. Parallel and Distributed Systems*, vol. 2, no. 2, pp. 242-252, April 1991.

[4] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Boston, MA, 1993.

[5] Y. Bar-Shalom and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, Storrs, (CT 06269-3157), 1995.

[6] D. Bertsekas, *Linear Network Optimization: Algorithms and Codes*, MIT Press, Cambridge, MA, 1991.

[7] H. Blom and Y. Bar-Shalom, "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients", *IEEE Trans. Automatic Control*, vol. 33, no. 8, pp. 780-783, August 1988.

[8] E. Daeipour, Y. Bar-Shalom and X. Li, "Adaptive beam pointing control of a phased array radar using an IMM estimator", in *Proc. 1994 American Control Conf.*, pp. 2093-2097, Baltimore, MD, June 1994.

[9] X. Li and Y. Bar-Shalom, "Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking", *IEEE Trans. Control Systems Technology*, vol. 1, no. 3, pp. 186-194, September 1993, Special issue on Air Traffic Control.

[10] X. Li and Y. Bar-Shalom, "Mode-Set Adaptation in Multiple Model Approach to Hybrid State Estimation", *Proc. American Control Conference*, Chicago, IL., pp. 1794-1799, June 1992; to appear in *IEEE Trans. Automatic Control*, 1996.

[11] K. Pattipati, S. Deb, Y. Bar-Shalom and R. Washburn, "A New Relaxation Algorithm and Passive Sensor Data Association", *IEEE Trans. Automatic Control*, vol. 37, no. 2, pp. 197-213, February 1992.

[12] K. Pattipati, T. Kurien, R. Lee and P. Luh, "On Mapping a Tracking Algorithm Onto Parallel Processors", *IEEE Trans. Aerosp. and Electron. Systems*, vol. 26, no. 5, pp. 774-791, September 1990.

[13] Popp, R., Pattipati, K., R., Bar-Shalom, Y., and Yeddanapudi, M., "The Parallelization of a Large-Scale IMM-based Multitarget Tracking Algorithm", *Proc. SPIE Conf. on Signal & Data Processing of Small Targets*, San Diego, CA, July 1995.

[14] M. Yeddanapudi, Y. Bar-Shalom, K. Pattipati and R. Gassner, "MATSurv: Multisensor Air Traffic Surveillance System", to appear in *Proc. SPIE Conf. on Signal & Data Processing of Small Targets*, San Diego, CA, July 1995.

[15] M. Yeddanapudi, Y. Bar-Shalom and K. Pattipati, "IMM Estimation for Multitarget-Multisensor Air Traffic Surveillance", to appear in *IEEE Trans. Aerosp. and Electron. Systems*, 1996.

C/C++: Utilized For
Data Monitoring and
Control

Peter A. Renzi
Graduate Student
Boston University

Boston University
Commonwealth Avenue
Boston, Massachusettes

Final Report:
Graduate Student Research Prog.
Rome Labs

August 1995

The idea behind this report is for us to look at some different aspects of C++ and build on them. The emphasis here is an actual program example. There is good reason for us to explore the ideas behind the program also. The text will be kept to a minimum, and mainly cover topics about things such as what the program does, how it was built, and its unique features.

OBJECT ORIENTED PROGRAMMING

Our job involves working as part of a team developing what is called a control system program for the production of a new enzyme.

Object oriented programming concentrates on taking a thing (like a fermenter) and breaking it down into objects (like temperature level, pressure level, flow level) and assigning methods and attributes (like lists of setpoints, tuning parameters, gain , reset , rate, agitation speed) to each object.

The temperature level has the ability of being changed and controlled. This means it needs to keep track of what the temperature is, what the tuning parameters of the controller are, and checking to see if if deviating from setpoint by too great a value. It also has attributes consisting of other objects - such as temperature report object , supervisory object, cascade object, remote/local object).

The temperature report object has the ability to store several transactions or cycles of time. It has the ability to print or store to a file.

There is also a batch report object. Each batch report has an associated product. As we make product, we assign setpoints, gains , reset levels, rate levels. There is also stuff common to all batch reports: our own company, production validation requirements, and verification procedures. We probably wouldn't want this information stored on a computer since it is too easy to steal. After a batch is completed the

information is passed on to the temperature , flow and pressure report object where it is recorded.

1. First the Batch Report Class:

Attributes: Temperature Report
              Flow Report
              Pressure Report
              Temperature Level
              Pressure Level
              Flow Level
              Setpoints
              Tuning Parameters
              Agitation Speed
              Disolved Oxygen Level

Methods: Print out report
              Save to disk
              Download information to flow , temp., and
              pressure report

2. Next, the Temperature, Flow, and Pressure Report Class:

Attributes: Current Value
              Changes

Methods: Print out report
              Save to disk

Change update time

3. And finally, the Fermenter Class:

Attributes: Temperature Setpoint, gain, reset, rate

                Pressure   "   "

                Flow   "   "

                Tag

                Another Fermenter

Methods: Change Initial Parameters

                Change running parameters

There can also be defaults in a parameter list for a method. Take a peek at the constructor in the code, and you'll see that there are = signs in it. That means if we don't include that parameter, it defaults to the value provided. The main() demonstrates this. Note that if we provide a default for one parameter, the parameters that follow must also have defaults, and if we use one of the defaults we must use the defaults for the rest of the parameters that follow. For instance, if we're using a method that has four parameters, all with defaults, and we only give two explicit parameters, those two go with the first two parameters in the method, and the remaining two get the defaults.

Often when you are modeling objects, you will want to provide default values in the methods. For instance, if you are writing a program that tracks customers' phone numbers and addresses, you may have a method called Save. Save may provide a default file name such as PHONES.TXT.

Passing to a Function a Pointer to a Class:

The function "tweak" takes a pointer to a Fermenter, and adds one to the Fermenter's gain. That's how the flow controller gets a change in a tuning parameter.

The base class is called Fermenter. It contains one fermenter. The variables that we are controlling are flow, temperature, pressure, motor speed, and dissolved oxygen. We can change the gain of the flow controller. The program also monitors how much product we have made and how much time the system was offline.

The tuning parameters are set in remote controllers. Our program uses methods to retrieve these variables and display them for the operator. We have set up the program to allow the operator to change the gain in the flow controller.

The program also reads data that was stored in a remote data logger. The system's logger is read by the DailyReport object. It contains three attributes online, special, and offline.

We have derived a fermenter1 class from the base class Fermenter. It can access all the values of the base class plus provides values for Dissolved Oxygen and MotorSpeed. The object is called agitation. Again the parameters for motorspeed and Dissolved Oxygen are set at the process level. The values are monitored at the computer. Since agitation is dervied from the base class, we can also monitor the tuning parameters within this class.

The key to inheritance and this program is knowing when to set/get values. For example you set values in the constructor or use default values. Thus you can overide the local controller in two ways - through the use of constructors or the set method. This program just utilizes the former. In fact if the values are set at the process level it is a good idea to use the default values.

When fermenter1 is created, the values change for tuning, dissolved oxygen, and motorspeed.

Class SetpointList is an efficient way to utililize a list to do batch processing. By batch processing set all control values in the process initially before the run.

DOT12F.H

```
#include <string.h>

enum tagcolors { red, green, blue, yellow, pink, silver };

class Fermenter
{

        char tag[20];
        tagcolors color;
        int gain;
        int reset;
        int rate;

public:

        Fermenter(char *theTag = " Pressure ",
        tagcolors theColor = red,
        int theGain = 5,
        int theReset = 50,
        int theRate =0)
```

```
{
strncpy(tag, theTag, 20);
color = theColor;
gain = theGain;
reset = theReset;
rate = theRate;
}

// functions for setting
void SetTag(char *theTag)
{ strncpy(tag, theTag, 20); }

void SetColor(tagcolors theColor)
{ color = theColor; }

void SetGain(int theGain)
{ gain = theGain; }

void SetReset(int theReset)
{ reset = theReset; }

void SetRate(int theRate)
{ rate = theRate; }

// functions for getting
void GetTag(char *buffer)
{ strncpy (buffer, tag, 20); }

tagcolors GetColor()
{ return color; }
```

```cpp
        int GetGain()
        { return gain; }


        int GetReset()
        { return reset; }


        int GetRate()
        { return rate; }


};


class Activity
{
public:

        // To make life easier, we'll make it  all  public.  Some
        // would say it's okay, anyway.

        Fermenter *ourFermenter ;
        int Online;         // How much product was made today.
        int Special;        // How much special product today.
        int Offline;        // How much down-time today.

        Activity (Fermenter *which);
        void SetStuff(int Online = 10, int Special = 10,
                int Offline = 10);
        void ReadStuff();
        void WriteStuff();
};
```

ACTIVITY IF.CPP

```cpp
// Here we use  the  alternate method of showing the
// code for the methods.


#include "dot12F.h"
#include <fstream.h>


Activity :: Activity (Fermenter *which)
{
        ourFermenter = which;

}
// We don't list  the defaults  for the  parameters here
// they only belong  in the  class  definition.


void Activity :: ReadStuff()
{
        // often, we just use 'f' for the file variable.


        char name[20];
        ourFermenter->GetTag(name);
        name[5] = '\0';                    // just use the first four
                                           // characters in the name.
                                           // Gotta have a buffer of
                                           // size 20, though, since the
                                           // GetName function will write
                                           // 20 no matter what.


        ifstream f(name);
        f >> Online;
```

```
            f >> Special;

        f >> Offline;

}


void  Activity :: WriteStuff()

{

        // often we just use 'f' for the file variable.


        char name[20];
        ourFermenter->GetTag(name);
        name[5] = '\0';


        ofstream f(name);
        f << Online << endl;
        f << Special << endl;
        f << Offline << endl;

}


void Activity :: SetStuff(int theOnline, int theSpecial,
                                int theOffline)


{
Online = theOnline;
Special = theSpecial;
Offline = theOffline;
}
```

TWEAKF.CPP

```
#include "dot12F.h"
```

```
#include "tweakF.h"


void tweak(Fermenter *which)
{
int gain;


gain = which->GetGain();
which->SetGain(gain + 1);
}
```

XF.H

```
void tweak(Fermenter *which);


// This control system adjusts for agitation


class Fermenter1 : public Fermenter
{
    int MotorSpeed;
    int DO;
public:


Fermenter1(char *Fermenter1Tag, int FermenterSize ,
    int InitMotorSpeed, int InitDO) : Fermenter(Fermenter1Tag,
    silver, 10,10,10)


{
    MotorSpeed = InitMotorSpeed;
    DO = InitDO;
}
GetMotorSpeed()
```

```
{return  MotorSpeed;}
GetDO()
{return DO;}
};
```

LIST.H

```
#include <string.h>
#include <assert.h>
#include <iostream.h>
#include <iomanip.h>
enum list10 {PressureController, TemperatureController,
FlowController};


class SetpointList {


public:
      const SetpointList *GetSetpointList() const;
      const GetSetpointChoice() const;
      int Setpoint1;
      SetpointList *Next   ;
};


class MyControlSystem {


public:


      SetpointList *Setpoints;
};


class Run: public SetpointList{
```

```cpp
public:

    Run(SetpointList *ListPtr, int setpointval);
};


Run::Run(SetpointList *ListPtr,int setpointval){

while(ListPtr){
    // Print the controller tagname with setpoint
    switch (ListPtr->Setpoint1){
        case PressureController:
            cout<<"PressureController"<<setpointval<<endl;


            break;
        case TemperatureController:

cout<<"TemperatureController"<<setpointval<<endl;
            break;
        case FlowController:
            cout<< "FlowController"<<setpointval<<endl;
}

ListPtr = ListPtr->Next;
}
}

int GetSetpointChoice(){
    int SetpointChoice;
    cout<<"What controller?"<<endl
    <<"1 = PressureController"<<endl
```

```cpp
            <<"2 = TemperatureController"<<endl
            <<"3 = FlowController"<<endl
            <<"0 = stop"<<endl;
        cin >>SetpointChoice;
// return this value.
return SetpointChoice;
}


SetpointList *GetSetpointList(){
        int SetpointChoice;
        SetpointList *ListPtr, *LastPtr=0, *First=0;
        while (SetpointChoice = GetSetpointChoice()){
        // Create a new list item for storing info.
 ListPtr = new SetpointList;
        // Store the  type.
        ListPtr->Setpoint1=SetpointChoice;
        // Clear the next pointer because there is  nothing
        // that  follows.
        ListPtr->Next = 0;
        switch(SetpointChoice){
        case PressureController:
            break;
        case  TemperatureController:
            break;
        case FlowController:
            break;
        default:
            cout<<"Pardon me?"<<endl;
            } // End of switch.
// Add the new item into the linked list.
```

```cpp
    if (LastPtr)

        LastPtr->Next=ListPtr;

    else

        First=ListPtr;


    // Update LastPtr.

    LastPtr=ListPtr;

}       // end of while statements.

// Return a pointer to the list.

return First;

}
```

DOT13F1.CPP

```cpp
#include "dot12F.h"

#include  "xF.h"

#include <iostream.h>

#include "tweakF.h"

#include "list.h"

void main()

{

    MyControlSystem good;

    good.Setpoints =  GetSetpointList();

    Run b(good.Setpoints, 20);


    Fermenter controller1;

    Fermenter controller2("Temp", green, 10, 100, 0);

    Fermenter controller3("Flow", blue, 5,5,1);

    char buffer[20];


    cout << "controller1   :" << endl;
```

```
controller1.GetTag(buffer);

cout<< buffer << endl;

cout<< controller1.GetColor() << endl;

cout<< controller1.GetGain() << endl;

cout<< controller1.GetReset() << endl;

cout<< controller1.GetRate() << endl << endl;


cout << "controller2:" << endl;

controller2.GetTag(buffer);

cout << buffer << endl;

cout << controller2.GetColor() <<endl;

cout << controller2.GetGain() << endl;

cout << controller2.GetReset() <<endl;

cout << controller2.GetRate() << endl << endl;


cout << "controller3:" << endl;

controller3.GetTag(buffer);

cout << buffer << endl;

cout << controller3.GetColor() <<endl;

cout << controller3.GetGain() << endl;

cout << controller3.GetReset() <<endl;

cout << controller3.GetRate() << endl << endl;


tweak(&controller3);

cout << "After tweak..." <<endl;

cout << controller3.GetGain() <<endl;


Fermenter Totalizer("Product");  // The remaining parameters
                                 // get the defaults.
```

```cpp
Activity DailyReport(&Totalizer);
DailyReport.SetStuff(5,4,3);
cout << "Initially:" << endl;
cout << DailyReport.Online <<endl;


cout << DailyReport.Special<<endl;
cout << DailyReport.Offline<<endl;
DailyReport.WriteStuff();
cout << "Saved these  values!" << endl << endl;


DailyReport.SetStuff(1,2,1);
cout << "New Values:" <<endl;
cout << DailyReport.Online <<endl;
cout << DailyReport.Special<<endl;
cout << DailyReport.Offline<<endl<<endl;


DailyReport.ReadStuff();
cout << "Read in previously saved values:" << endl;
cout <<  DailyReport.Online << endl;
cout <<  DailyReport.Special << endl;
cout <<  DailyReport.Offline << endl << endl;


// Fermenter I - Agitation
Fermenter1 agitation("Agitation",40,10,1000);
cout << agitation.GetMotorSpeed();
cout << agitation.GetDO();
cout << "Finally let's demonstrate  inheritance" <<endl;
cout << agitation.GetGain();
cout << endl;
```

}

PROGRAM OUTPUT

ug 30 10:42:03 EDT 1995

Chiron> doggy90

What controller?

1 = PressureController

2 = TemperatureController

3 = FlowController

0 = stop

2

What controller?

1 = PressureController

2 = TemperatureController

3 = FlowController

0 = stop

1

What controller?

1 = PressureController

2 = TemperatureController

3 = FlowController

0 = stop

2

What controller?

1 = PressureController

2 = TemperatureController

3 = FlowController

0 = stop

0

FlowController20

TemperatureController20

FlowController20

controller1 :

Pressure

0

5

50

0

controller2:

Temp

1

10

100

0

controller3:

Flow

2

5

5

1

After tweak...

6

Initially:

5

4

3

Saved these values!

New Values:

1

2

1

Read in previously saved values:

5

4

3

101000Finally let's demonstrate inheritance

10

OPTICAL CURRENT CONTROLLED OSCILLATORS
FOR MICROWAVE COMMUNICATIONS

David H. Sackett
Department of Electrical Engineering

Rochester Institute of Technology
One Lomb Memorial Drive
Rochester, New York, 14623

11-1

Optical Current Controlled Oscillators For Microwave Communications

David H. Sackett
Department of Electrical Engineering
Rochester Institute of Technology

Abstract

Most available Laser Diodes(laser diode) display varying degrees of Self-Sustained Pulsation(SSP). SSP, TSP(Transient-Sustained Pulsation), and FSP(Feedback-Sustained Pulsation) are forms of Optical Current Controlled Oscillators. The characteristics of SSP were explored and optimized to create a communication system utilizing SSP as a microwave sub-carrier upon an optical carrier. A feedback system and filtering were used to stabilize the SSP and TSP resulting in an FSP communication system. The FSP communication system was unsuccessful in producing reliable bit error rates due largely to carrier harmonics, non-linear modulation, and excessive noise.

Table of Contents

Table of Figures

I. Introduction

_____ *Self-Sustained Pulsation(SSP)*

A light emitting diode(LED) is a p-n junction in which the band-gap energy corresponds to an electromagnetic photon energy ($E=h\omega$). When electron-hole pairs recombine an electromagnetic photon is emitted. By creating population inversion (more gain than loss) and utilizing stimulated emission of photons (trapping the photons in a mirrored cavity), a laser diode(laser diode) is created. The process of stimulated emission of photons creates a coherent laser light output from the laser diode.

Unlike traditional solid, liquid, and gas lasers a laser diode creates a population inversion by applying a voltage across the p-n junction. A laser diode requires population inversion in order to create gain and therefore emit sufficient quantities of coherent light. The current at which the laser diode gain equals the loss(absorption) is the threshold of laser diode. Increased current above the threshold results in lasing and laser diode output... The laser diode is on. See Figure A.
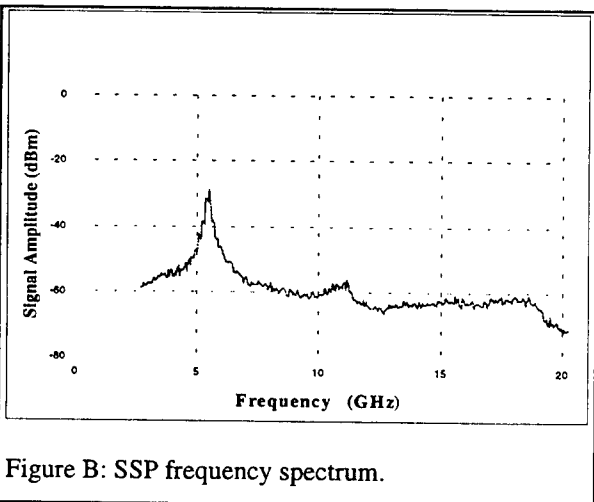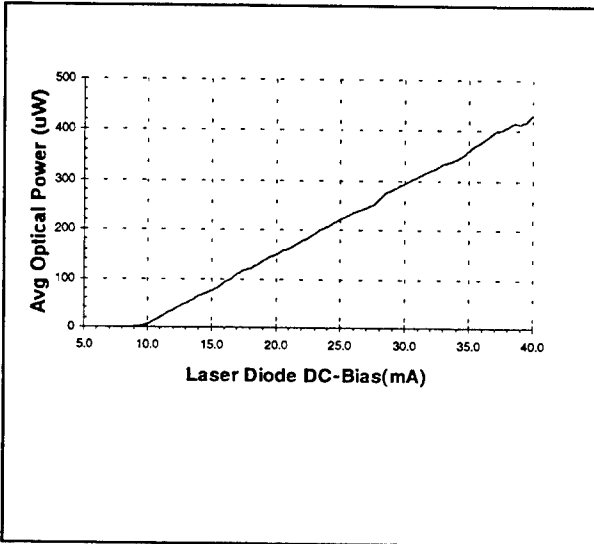
Figure B: SSP frequency spectrum.

Most Laser diodes emit light power from one to ten milliwatts. This light is usually continuous wave(CW) and only varies slowly with time. It was discovered with high-speed detectors that most laser diodes actually have very high frequency intensity variation; on the order of gigahertz. This SSP was unintentional, but was oscillating so rapidly and weakly that it was of no concern to the then slow laser diode applications. See Figure B. The signal of Figure B can be described by the power Equation #1:

$$P_{SSP} = 204\mu W + 135pW \cos(\omega_c t) + 4.28pW \cos(2) \qquad (1)$$

The SSP frequency varies with the LD DC-bias current. This trend is shown is Figure C. The measured data was fit to a second order polynomial and a very good approximation to the data is

$$SSP\ frequency = -006848x^2 + 0.4778x - 2863 \qquad (2)$$

The fact that the SSP frequency is variable leads to the idea of frequency modulating this microwave sub-carrier. Frequency modulation lends itself to frequency multiplexing and FSK. Frequency modulation systems usually display superior SNR when compared to the equivalent amplitude modulation of the system.

Figure C: SSP frequency variation with LD DC-bias current.

The amplitude of SSP is very small and is usually negligible when compared to the CW component. Figure B shows the microwave spectrum of a typical 1300nm SSP laser diode. Oscillation with a fundamental at 5.5 GHz is very wide, on the order of 500 MHz, and no harmonics are visible. Some SSP Laser diodes have as narrow as several megahertz and enough power to display one or two harmonics. Because different manufacturing processes vary the absorption and emission parameters the strength of SSP in a particular laser diode varies greatly with wavelength and laser diode architecture.

11-5
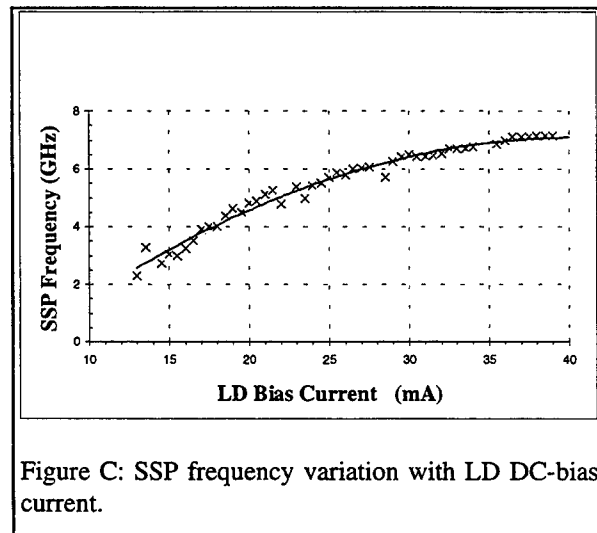
SSP can be interpreted as a millimeter(gigahertz) intensity modulation placed on an optical carrier. Such modulation was originally considered a problem and endeavors were begun to remove the impurities contributing to the competing absorption and emission characteristics. It was quickly discovered that the presence of SSP reduced the Relative Intensity Noise(RIN) and can also reduce distortion due to aberration.

The SSP is modeled as the result of two cavity regions with different absorption and emission characteristics. The photons travel through both regions with each pass across the optical cavity. The different photon lifetimes of the two regions create an oscillating equilibrium. An excess of photons causes a depletion of photons which then causes the original excess of photons.

Several efforts have actively enhanced the SSP phenomenon for low noise applications. Such laser diodes are used for high density Compact-Disk(CD) storage devices and communication systems. The application dictates what characteristics are desirable, but for communication systems it is desirable for the SSP spectrum profile to have a narrow line-width, deep intensity modulation index, and single frequency component (no harmonics).

Transient-Self Pulsation (TSP)

There are bias regions where the SSP is not stable. The SSP will quickly reduce until it is gone. The laser diode is exhibiting Transient-Self Pulsation(TSP) in these regions. For some laser diodes the TSP regions are wider than the SSP, for others the opposite holds true. The border between these regions is not rigid and is history dependent. TSP appears as SSP when the current is first tuned to that TSP location or when the laser diode is turned on at a TSP bias current. The TSP then reduces in power until it is gone. When biased very close to an SSP/TSP border TSP will some times reduce and then SSP will rise close by in an SSP region.

SSP can be considered a tunable optical sub-carrier. The DC bias current is tuned to se-
lect a sub-carrier frequency. This optical micro-
wave amplitude(intensity) modulation is simply a
tone signal being carried on an optical carrier.
The sub-carrier is unsatisfactory because of its
large line-width, variance in characteristics from
laser diode to laser diode, and low modulation
index. All of these parameters are improved with
the use of Feedback-Sustained Pulsation(FSP).
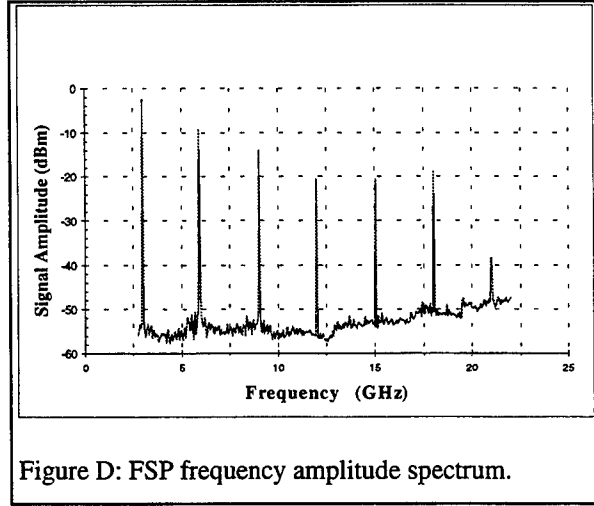
Figure D: FSP frequency amplitude spectrum.

FSP is created by feeding back some of the electrical signal from a photodiode to the DC-
bias for the laser diode. The feedback changes the charge density in the laser diode across the
SSP frequency range. The periodic presence of excess charges to be recombined causes an
equally periodic presence of excess photons. These photons are then detected and fed back to the
laser diode charge density. The process continues reducing the line-width with each cycle. The
line-width stabilizes at 10 to 50 kilohertz; very narrow when compared to the 10 MHz to 1 GHz
line-width of SSP. Figure D can be expressed mathematically by the following equation:

$$P_{FSP} = 209\mu W + 92.10\mu W \cos(\omega_c t) + 42.19\mu W \cos(2\omega_c t) + 24.55\mu W \cos(3\omega_c t) + 11.35\mu W \cos(4\omega_c t) + \\ + 11.35\mu W \cos(5\omega_c t) + 14.29\mu W \cos(6\omega_c t) + 1.50\mu W \cos(7\omega_c t) \tag{3}$$

Traditional laser diode communications systems employ Base-Band Amplitude Modulation(AM). The information signal is add to the DC-bias of the laser diode and the intensity increases with the electrical signal power. So long as the electrical signal remains above the laser diode threshold the relationship between the signal and the laser diode CW power output is linear. This relationship is shown in Figure A on page 5. Thus if the laser diode is biased in the middle of the current range the electrical signal will be intensity modulated onto the optical carrier. For such a communication system any microwave sub-carriers must be created electrically and all signals must be combined before applying the signal to the laser diode. Thus the optical system is not providing any additional bandwidth because the information had to be constructed electrically first and only then placed on the optical carrier. The only benefit attained is that of transmission. Copper wire is considered to provide good conduction, but the attenuation over long distances at high frequencies requires high power levels and thus heavy cable. The extremely low loss of optical fiber (~0.2 dB/km) allows very low power (~milliwatts) over far distances with very little cost. SSP is ignored in such a system because its modulation index is very small. SSP is a noise term to be filtered in these systems.

FSP and strong SSP may also be amplitude modulated. But in this case only the base-band signal is needed. The microwave carrier system is generated within the laser diode. Thus a low bandwidth system may be directly placed onto a microwave carrier which may be much higher than the bandwidth of the electrical system. In addition, the SSP optical carrier's frequency is tunable allowing the possibility of using several microwave sub-carriers on the same optical carrier. This amounts to frequency multiplexing in the microwave frequency range. Frequency multiplexing greatly increases the transmission system bandwidth above the individual systems' bandwidths. A communication system utilizing the SSP phenomenon would therefore dramatically increase transmission system bandwidth with virtually no additional cost!

The frequency of the microwave sub-carrier present in an SSP laser diode changes with the laser diode DC-bias current. See Figure B on page 4. This phenomenon provides the ability to space out the microwave sub-carriers. However, when the DC-bias is varied quickly, such as when a modulation signal is added to the laser diode DC-bias current, this signal varies the sub-carrier frequency quickly producing Frequency Modulation(FM). FM has superior characteristics compared to AM for communication systems. FM is less subject to interfering noise and has better Signal-to-Noise(SNR) characteristics due to noise suppression.

The microwave sub-carrier may be Amplitude and Frequency modulated by varying the bias current with an information signal. Typically the laser diode will respond to killohertz through Gigahertz of information modulation. These information signals may be sub-carriers for even lower frequency data.

II. The Experimental Setup

De-Modulator

Tektronix 11801
Oscilloscope

HP8657B
Microwave
Spectrum Analyser

Input
Signal
S(t)

-6dB

+20dB

90 Deg
-3 dB

Δ t

Mixer

Output
Signal
O(t)

-6dB

Attenuator

BPF

-6dB

+20dB

COM

LD

PD

+20dB

Idc

Anritsu MS9001B1
Optical Spectrum
Analyser
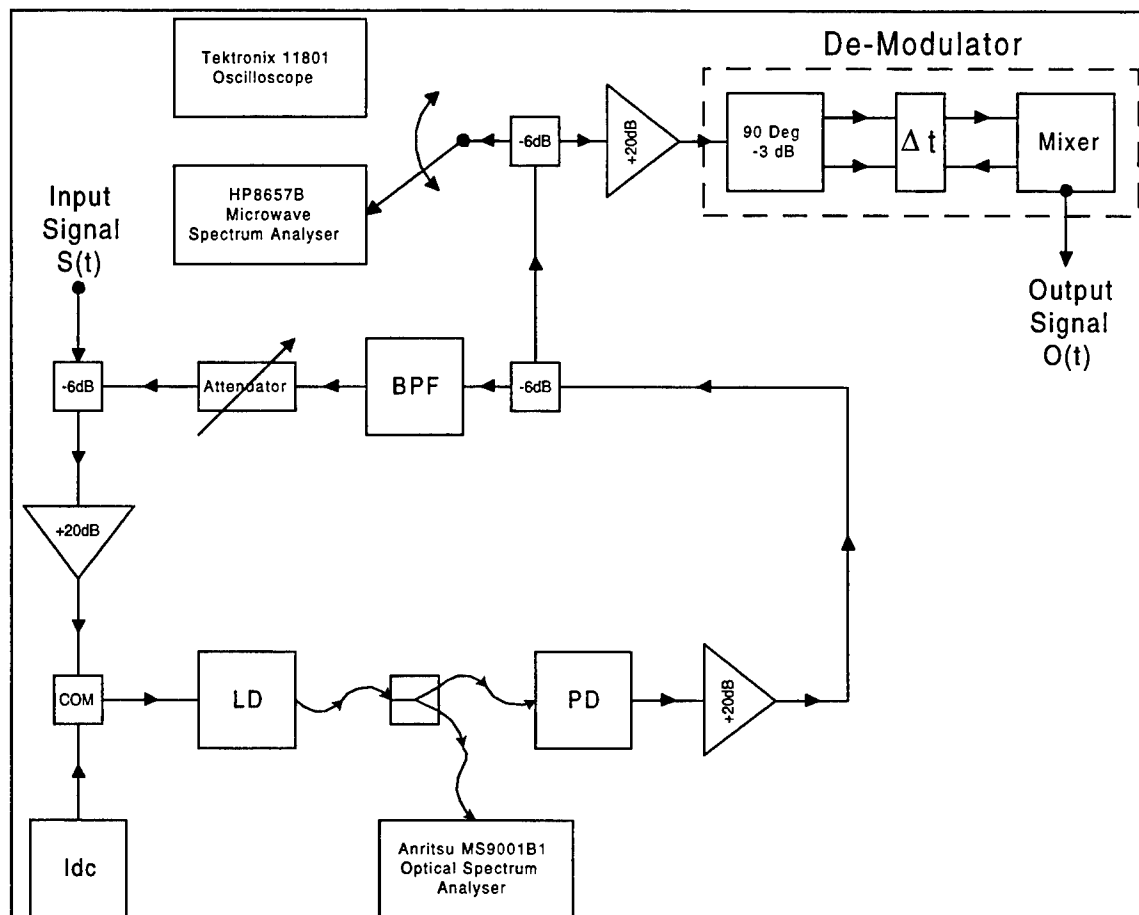
Equipment List:

Input Signal: Hewlett Packard HP8657B
Microspectrum Specturm Analyser: Hewlett Packard HP8593E
Oscilloscope: Tektronix 11801
    Trigger Head: SD-51
    Detector Head: SD-26
Optical Analyser: Anritsu MS9001B1
LD Controller: SeaStar TC-5000
Photodector: New Focus 1414

Laser Diode: Mitsubishi ML7011R
Optical Power Meter: Newport 835
90 Degree Shifer: Mini-Circuits ZAPDQ-4
Mixer: Mini-Circuits ZEM-4300MH
Band Pass Filter: K&L 4FCT-2000/4000-1-O/O
Power Divider: Hewlett Packard HP11667B
Variable Attenuator: Hewlett Packard HP8494B
Amplifiers: Miteq AMF-40-00

Figure E: FSP communication system diagram.

The feedback power must be adjusted with the variable attenuator seen in Figure E. Too little feedback will result in SSP and too much produces FSP collapse. Between the FSP and SSP is another region of FSP collapse. FSP threshold information can be seen in Figure F.

Figure F: SSP/FSP threshold. The needed feedback power increases with LD DC-bias current.
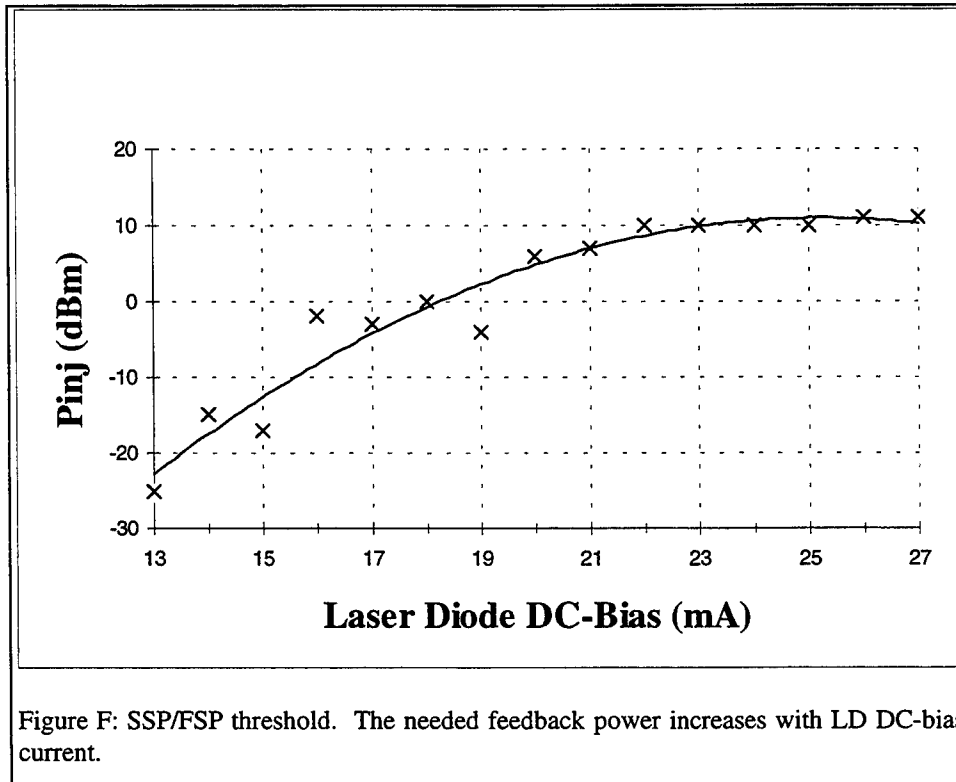
For the laserdiode used in this experiment it was not possible to increase the feedback high enough to cause the second region of FSP Collapse. The line in Figure F shows the second order regression values of the threshold level. As the laser diode DC-bias current is increased the FSP threshold also increases; and it increases at a rate which is non-linear to the laser diode DC-bias current. In Figure F the region below the threshold line represents a system output of FSP. The region below the line signifies insufficient feedback levels and results in SSP output. At the threshold is observable FSP collapse. FSP collapse manifests in a broad region of narrow spikes which are chaotic in amplitude and frequency. The amplitudes are well below the FSP signal level amplitude, but tend to be greater than the SSP amplitude.

For this communications application the harmonics created during FSP are undesirable. To more clearly focus the feedback affects the feedback loop contains a band-pass filter(BPF) as seen in Figure E. This narrow band filter provides excellent harmonic suppression. Figure G shows the signal after filtering which is feedback into the laser diode.
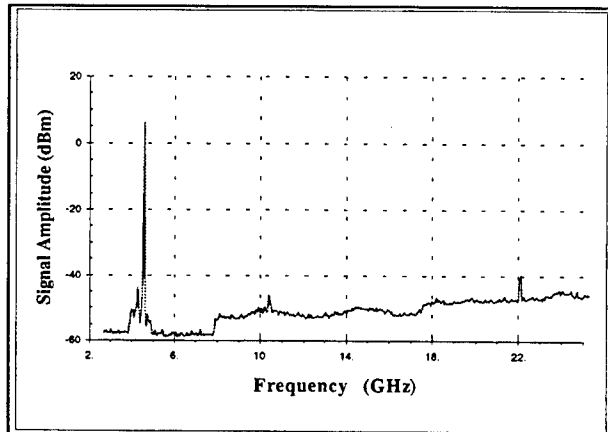
Figure G: Feedback signal after the band-pass filter.

III. Results

The FSP harmonics are an obstacle to frequency division multiplexing. By varying the LD DC-bias current and the feedback injection power the harmonics may be reduced. The cost is lower fundamental power. Although the harmonics are not gone, the spacing between each successive harmonic is increased. The fundamental may be amplified to any desired value, but the loss in SNR is not recoverable Figure H resulted when the LD was biased very close to the threshold current and the feedback injection power was minimized. Because of the low SNR
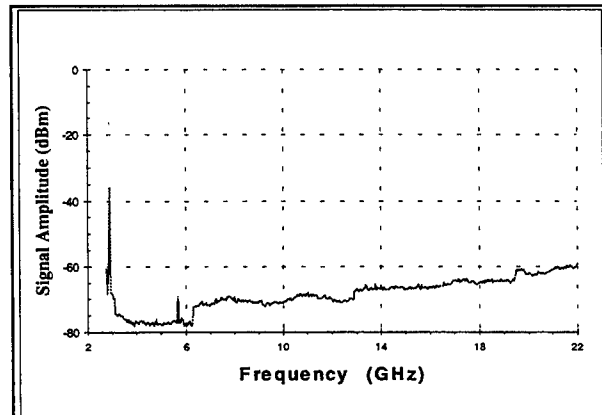
Figure H: Minimum FSP Harmonics at minimum feedback with Idc near LD threshold.

of the reduced harmonic signal the system testing was performed at DC-bias currents between 25mA and 30 mA. This produces several harmonics which were ignored or filtered as appropriate.

_____ *FSP Time Domain Analysis*

The SSP signal amplitude for the laser diode used in this experiment was too close to the noise floor level to be visible on an oscilloscope. The FSP signal amplitude, however, was large enough to provide stable traces on a high speed oscilloscope. A Tektronix 11801 40ghz digital oscilloscope was used to produce the time domain waveforms shown in Figure I. The FSP spectrum plots provided so far are amplitude plots only. The phase of the spectral components is not measured by the spectrum analyzer. Therefore it is not possible to extract the time domain information. A high speed oscilloscope or a network analyzer is necessary to obtain the time do-

main FSP response. Figure I shows three time plots at minimum, mid, and max DC-bias current into the laser diode.

The waveforms shown resemble pulses rather than sign waves due to the carrier harmonics which can be seen in Figure D. These harmonic component amplitudes are quantified in Equation 3. These results correspond to the simulated results based on the simple physical two-section model discussed previously. The simulations predict sharp pulses at a low duty cycle. The experimental waveforms display a duty cycle of approximately 30%. The actual duty cycle may be smaller still. The system is bandwidth limited to roughly 20GHz by components such as the amplifiers.

Figure I: FSP time response. a) Idc=14.0 mA b) Idc=25.0 mA c) Idc=40mA
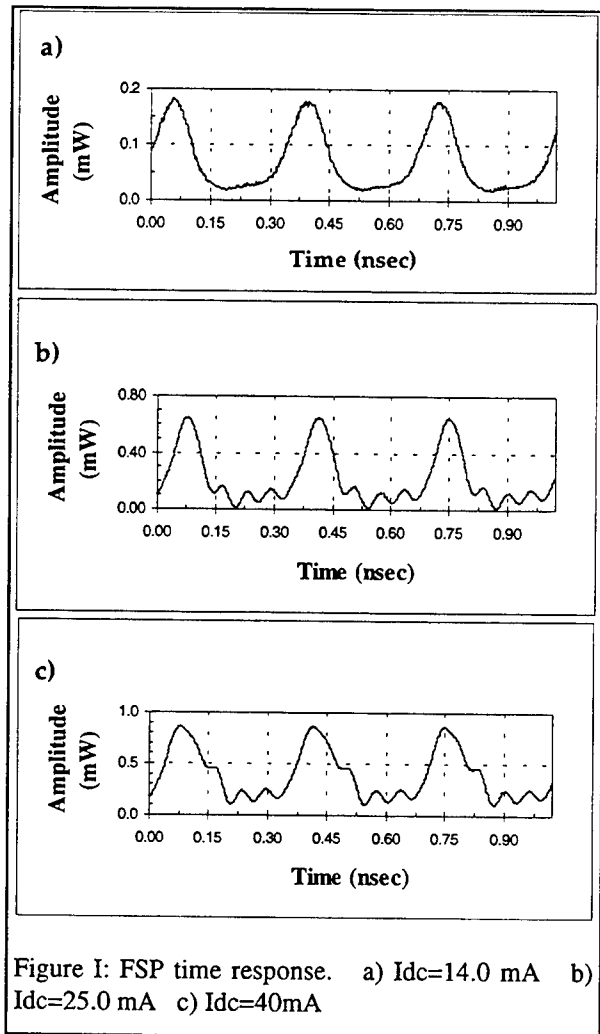
It was shown in Figure C that the SSP characteristic was tunable over 5GHz. It is desirable to have the same Tunability for FSP to provide wider band modulation. Figure J shows the limited Tunability of the laser diode used in this experiment. The Tunability characteristics are disappointing. The frequency is only tunable over 5MHz allowing only very narrow band frequency modulation. In addition the FSP amplitude varies widely over this entire range resulting in significant amplitude modulation on the carrier as well as the desired FM.

Figure J: FSP Tunability; a) Frequency b)Amplitude.
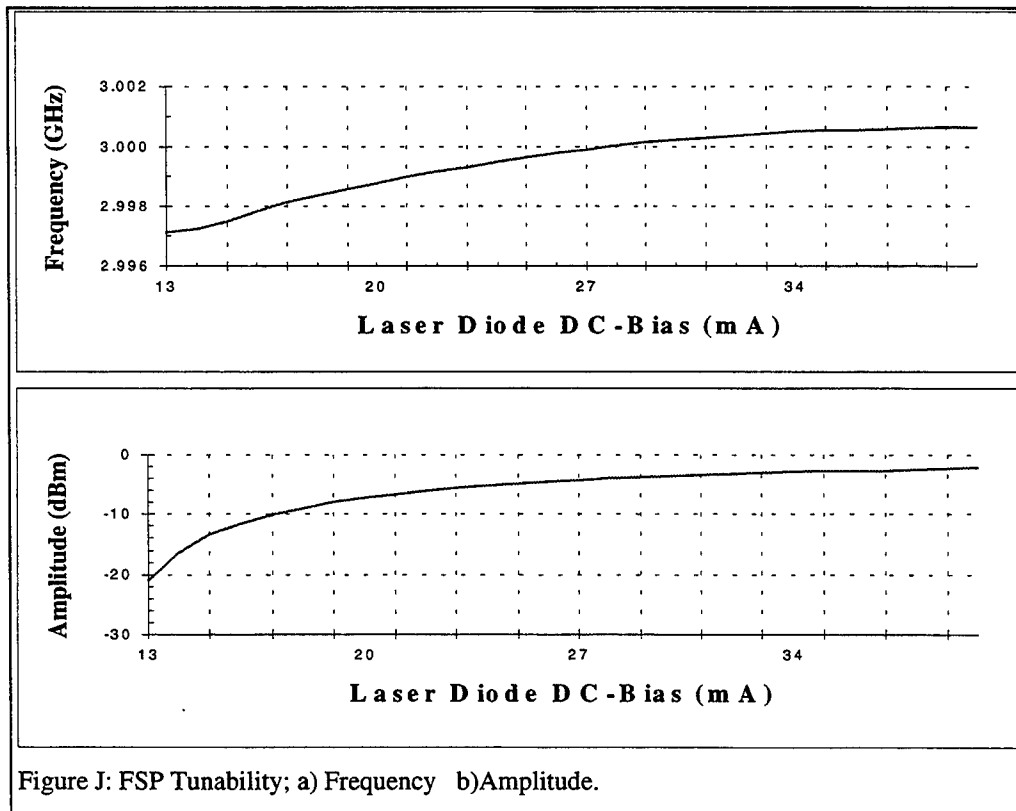
Placing a tone as the source input S(t) as shown in Figure E results in FM. Samples of the FM observed as shown in Figure K.
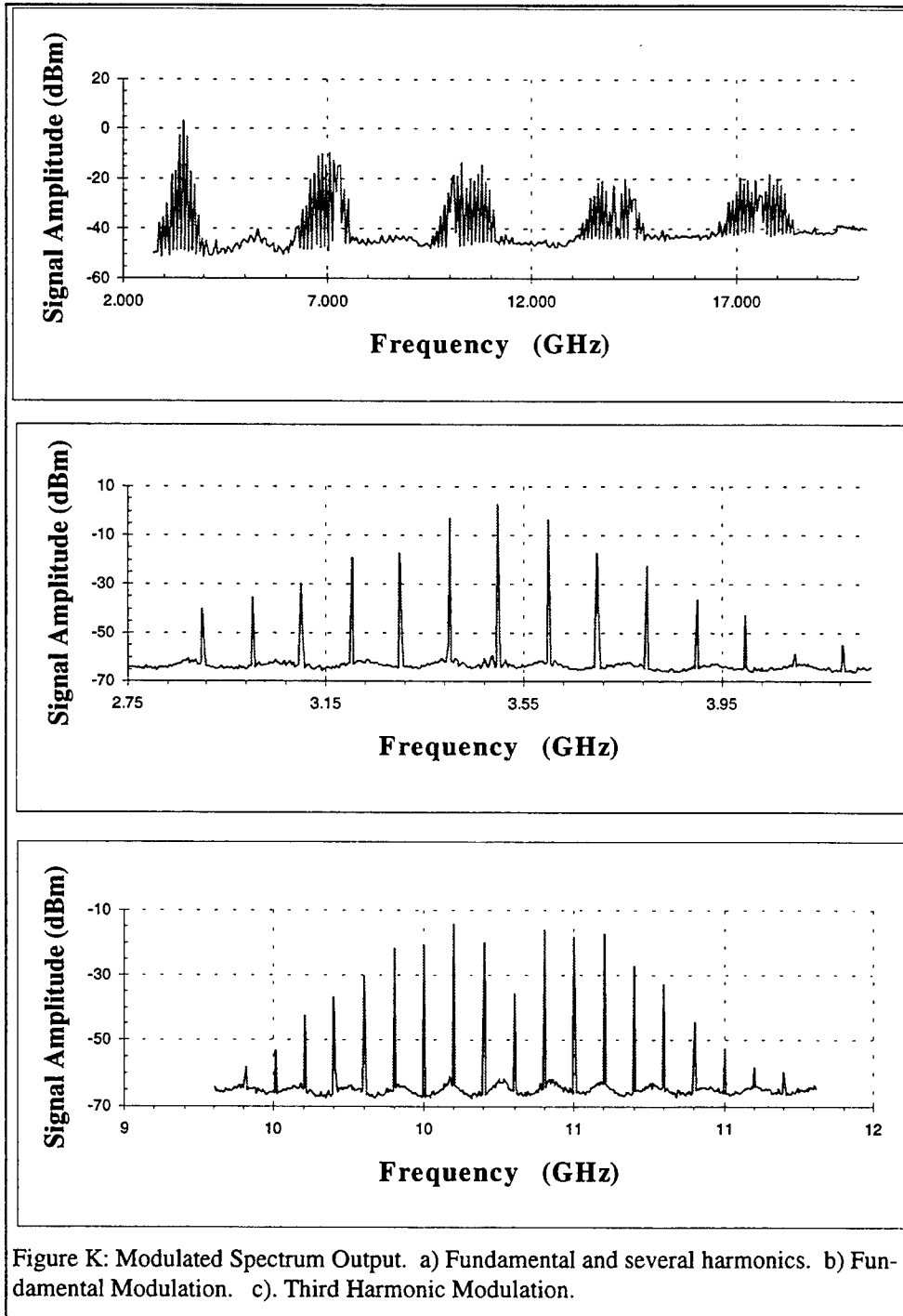
Figure K: Modulated Spectrum Output. a) Fundamental and several harmonics. b) Fundamental Modulation. c). Third Harmonic Modulation.

IV. Conclusions

In conclusion the experimental data gathered shows that the laser diode used in this experiment was not a good choice for this application. Desirable characteristics have now been identified by these experiments and include; strong SSP, smooth frequency tuning of SSP and FSP, Low optical harmonic generation, linear or constant FSP amplitude tunability, low level feedback threshold for sustained FSP, and removable of AM. Although no laser diode has yet to demonstrate all of these characteristics, previous laser diodes have demonstrated several of these characteristics. The laser diode used in this experiment demonstrated minimum SSP qualities.

Proper filtering of the output harmonics was not possible due to the lack of components. The extra bandwidth and signals made time domain analysis of the demodulated base-band signals impossible. As a result although SNR and dynamic range plots were taken with the spectrum analyzer, BER rate and actual data transfer attempts failed. In addition no output was stabilized on the oscilloscope to observe the phase distortion of the system.

A Study of Uncertainty in Image Processing

John F. Swan-Stone
Ph.D. Student
Systems Science Department

Watson School of Engineering
SUNY-Binghamton,
Binghamton, NY 13901

A Study of Uncertainty in Image Processing

John F. Swan-Stone
Ph.D. Student
Systems Science
SUNY-Binghamton, NY 13901

Abstract

First, An overview of the various forms of uncertainty is given followed by an argument as to why we need to recognize and manage uncertainty in today's models so as to be able to develop computationally feasible systems of organized complexity. Second, a brief description of the roles that uncertainty plays in the field of image processing is presented along with a special emphasis on fuzziness and image contrast enhancement. Lastly, a genetic algorithm is developed which is able to optimize a fuzzy contrast enhancement operator by minimizing the resulting uncertainty in an image. Experimental results are presented followed by conclusions and areas for further investigation.

Introduction

One of the goals of Rome Labs is the development of systems and models capable of managing uncertainty in a rational manner. Albert Einstein was one of the first people to recognize the need for managing uncertainty as can be seen in his 1921 declaration: *So far as laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality*. Great strides have been made in the study of uncertainty since the turn of the century, but it is apparent that we have only began to scratch the surface of this critically important field of study. However, we do know that there are essentially three distinct forms of uncertainty: *Non-Specificity, Probabilistic Uncertainty*, and *Fuzziness*.

Non-Specificity is uncertainty which occurs in situations where the alternatives are clear, but there isn't enough information to determine the correct one. For example, given the question "Should I take a plane, train, or automobile?", I have evidence that I should be a passenger <plane or train>. This evidence is non-specific because it doesn't enable me to decide which alternative to take. Another example is the guilt or innocence of a defendant. The alternatives are clear, guilty or not, but the true state of affairs can be very obscure (the O.J. Simpson trial is a good example) In general, the less specific the evidence in a given state of affairs, the higher the uncertainty.

Probabilistic Uncertainty is uncertainty in which conflicting claims are made regarding the outcome of a situation. For example, given the above question involving planes, trains, and automobiles, probability theory requires that I make precise and conflicting claims about the situation i.e. <Prob(Plane) = 0.3 vs. Prob(Train)=0.5 vs. Prob(Automobile)=0.2>. The evidence regarding the plane is in direct conflict with the evidence regarding the train and the evidence regarding the automobile. In general, the larger the conflict in evidence the larger the amount of uncertainty in the given situation. (NB: This is the only type of uncertainty capable of being captured in Probability Theory).

Once we have mathematical formalisms capable of capturing and measuring the magnitude of these forms of uncertainty, a concept of Information can be defined as the reduction in uncertainty of a situation by taking some action such as observation, experimentation, data collection, etc.

Fuzziness is concerned with the degree of compatibility of a measurable entity with an abstract concept, i.e. classification. Typical classification is done using sharp, distinct boundaries: For example consider the concept of Tall People. At what height should a person be considered tall? Some would say people who are 6 ft in height are tall people. But what of people 5 ft 11 and 15/16 inches? Are they not tall? Is it reasonable that an arbitrarily small difference in height from 6 ft justifies the classification of not tall? Yet this is exactly what classical classification techniques <logic/set theory> demand. Fuzziness allows the concept of degrees of compatibility. In the above example someone 5 ft 11 and 15/16 inch in height would be classified as someone who is tall to degree 0.99, someone 5 ft 6 inches would be tall to degree of perhaps 0.5, and someone 4 ft 0 inches would be compatible with the concept of tallness to degree 0.0. These numbers have to be determined, of course, in the context of each particular application. For example, the compatibility of a measured value temperature with the concept of "High Temperature" is very different in the contexts of medical diagnosis and nuclear reactor diagnosis. In general, the larger the region of partial compatibility, the larger the fuzziness.

At this point it should be mentioned that Fuzziness is not connected with the measurement of Information as are Non-Specificity and Probabilistic Uncertainty. Fuzziness is linguistic in nature and can be combined with the other two types of uncertainty which result from information deficiency.

There were scientists in the early part of this century who believed that if we could only make precise enough measurements and had sufficient computing power, we would be able to predict any future state of the universe with any desired precision. This view was critically examined by Hans Bremermann in 1962 who computed a theoretical upper bound on information processing using simple considerations of quantum theory. The limit is expressed by the proposition: "No data processing system, whether artificial or living, can process more than 2×10^{47} bits per second per gram of its mass." Bremermann then considered a hypothetical computer the size of the earth running since the beginning of the universe, and calculated that it would have processed approximately 10^{93} bits of information. This number is commonly referred to as *Bremermann's limit*, and problems which require more than 10^{93} bits of information are called *transcomputational problems*. This seems rather discouraging at first because it can be shown that even systems of modest size exceed the limit in information-processing demands. However, we can trade the requirement of precision for a reduction in complexity in our models. In other words, we can create models which contain uncertainty in return for computational feasibility. In general, allowing more uncertainty tends to reduce complexity and increase credibility of the resulting model.

Uncertainty in Image Processing

Probabilistic Uncertainty

The application of models capable of capturing various forms of uncertainty to the field of image processing has led to some very successful results. For example, the Maximum Entropy Restoration image enhancement operator [22] is capable of producing remarkably improved images in certain circumstances. This approach is based on assuming that each pixel in an image is related to the true physical image by the values of its neighbors and gaussian noise with mean zero, and unknown variance. Under some constraints regarding the distribution of the variance, and validity of the model (i.e. the satisfaction of Kolmogrov's axioms), a new image is created by maximizing the amount of Probabilistic Uncertainty in the original image. The amount of improvement in the new image is directly related to the validity of the underlying assumptions regarding the uncertainty present in the image, i.e. that of zero-mean gaussian noise, and the validity of the model relating the value of the pixels to the real world.

Non-Specificity

Another researcher, Mark Weirman[38] ,has extended the idea of Maximum Entropy reconstruction to the consideration of Non-Specificity in an image as captured in Possibility Theory. His results show that this type of reconstruction produces better solutions for some, but not all, of the cases where Probabilistic Maximum Entropy methods fail to do so.

Fuzziness

Fuzziness in pattern recognition

As to the application of fuzziness in image processing, a literature search has yielded papers which identify three distinct areas in fuzzy image processing (see bibliography). The first is concerned with the fuzzification of mathematical techniques used in pattern recognition, in particular the Fuzzified Hough Transform[17], and Hough Transforms with Fuzzy Probabilities[16]. These approaches are attempting to capture the uncertainty involved in the relationship of precise measurable entities, length, width, color, spatial connectivity, to abstract concepts such as wrench, screwdriver, airplane, tank, and, as such, seem to involve the fundamental issues which fuzziness was created to address. The Hough transform is a image processing technique at the highest level of the image processing hierarchy where the hierarchy consists of, from lowest to highest, image

acquisition and enhancement, feature extraction, and pattern recognition. Research into this approach would have been my first interest had it not been for the unacceptable investment in time and resources required to implement the lower level processing necessary before experimentation with fuzzified Hough transforms could begin.

Fuzziness in clustering techniques

The second area is a generalization of the c-means clustering algorithm to one of fuzzy c-means clustering. Clustering algorithms attempt to categorize the domain of interest into c unique partitions based on minimizing a measure of distance relating each element to every other element. Fuzzified c-means clustering assigns a degree of membership for each element in the domain of discourse to c fuzzy sets. In other words, each item can belong to more than one partition with varying degrees of membership. Again, the membership assignments are based on minimizing a measure of distance between the elements. This approach is useful for assigning the precise measurable quantities of color or gray level associated with each pixel to abstract concepts such as water, land, or man-made object. Unfortunately, this approach suffers from a very high computational complexity and in some instances, lack of information regarding the correct number of clusters present in an image, i.e. non-specificity.

Fuzziness in image contrast enhancement

The third area is concerned with image contrast enhancement using fuzzy techniques to represent the degree of compatibility the pixel gray-level has with the concepts "black" and "white". A low contrast image is one in which there is a high degree of uncertainty regarding this classification possibly due to an unwanted non-linear transformation in the processes involved in capturing the image. The idea of fuzzy contrast enhancement is to reduce this uncertainty and improve the image by using a non-linear "reversing" contrast enhancement operator.

Closely following the work De and Chatterji[1] and Fang and Cheng[2], an image, G, of dimension IxJ, and with L levels can be considered as an array of fuzzy singletons. Each value of membership, μ_{ij}, denotes the degree of having brightness, g_{ij}, relative to some level, L (g_{ij} = 0, 1, 2, ..., L-1). These membership values are positive and where μ_{ij} = 0.5, the gray level g_{ij} is said to be a crossover point dividing the "white" pixels from the "black" pixels. This is because when we talk about contrast, it is maximum when the values of the pixels are either 0 (black) or 1 (white), but when the gray-level of a pixel can take on values of a continuous gray-scale, we would like to divide the gray-scale into two regions, the black band and the white band. In other words, we want to classify the pixels as belonging to either "white" or "black" with the help the pixel's corresponding membership function. Therefore, we can say that "white" pixels are those whose membership is less than or equal to 0.5, and "black" pixels are those whose membership is greater than 0.5.

Before this classification can be made, it is first necessary to specify a fuzzification function. That is, a function which for every gray level assigns a membership grade, μ_{ij} specifying the degree to which that particular gray level is compatible with "black" and by using the definition of complement, "white". Taking advantage of the fact that when the gray levels of an image are tightly clustered around a value, say m, the contrast is very low and taking advantage of the fact that when the membership values are very close to 0.5, the fuzziness is at a maximum, we can propose a fuzzification function of the form.

$$\mu(g) = \frac{1}{2}\left(1 - \frac{m-g}{m-l_0}\right) \text{ for } l_0 \leq g \leq m,$$

$$\mu(g) = \frac{1}{2}\left(1 - \frac{g-m}{h_0-m}\right) \text{ for } m \leq g \leq h_0$$

where,

g = gray level of the original image,
h_0 = maximum gray level of the original image,
l_0 = minimum gray level of the original image,
m = subject threshold of the original image, decided by the user.
$\mu(g)$ = fuzzy membership of the gray level g.

This linear function transfers the two dimensional image data from the spatial domain to the fuzzy property domain, and allow us to stretch and compress different regions as can be seen in the following figure.

Gray Level of Original Image
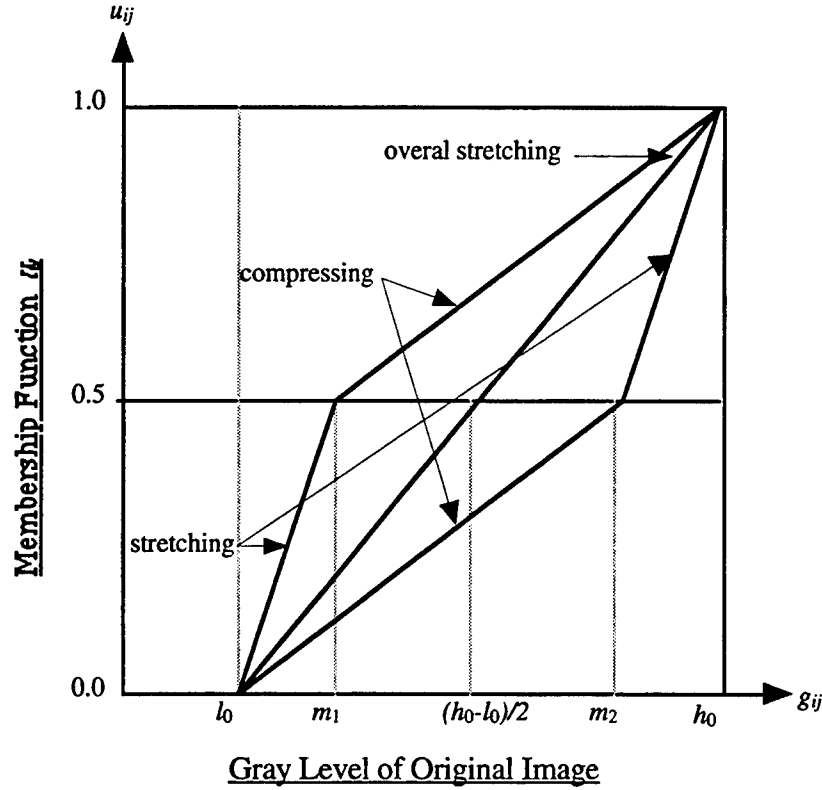
Once an contrast enhancement operator has been applied to the fuzzy property domain, it is necessary to apply an inverse function called the defuzzification operator, to produce the modified image. We will use the following function, which is the inverse of the fuzzification function.

$$g' = n - (n - l_e)(1 - 2\mu'(g)) \ \text{ for } 0 \le \mu'(g) \le \frac{1}{2},$$

$$g' = n + (h_e - n)(2\mu'(g) - 1) \ \text{ for } \frac{1}{2} \le \mu'(g) \le 1$$

where,
g' = gray level of the enhanced image,
he = maximum gray level of the enhanced image,
le = minimum gray level of the enhanced image,
n = subjective threshold of the enhanced image, decided by the user.

Here the new desired threshold n allows us to further emphasize some gray levels while de-emphasizing others.

Once we have transformed our image into the fuzzy property domain, we now wish to reduce the fuzziness to realize a corresponding increase in image contrast. Zadeh[36] has proposed the INT

contrast intensification operator. When applied to a fuzzy set it generates a new fuzzy set such that the gray levels are shifted apart following two piece wise quadratic functions. De and Chatterji [2] have proposed an alternative contrast enhancement operator which has several more free parameters and allows a flexible transformation which more closely models the nature of the human visual response characteristic. This operator is defined as:

$$\mu'(g) = \frac{1}{2}\left(1 - \left\{\frac{\sin\theta_1(1-2\mu(g))}{\sin\theta_1}\right\}^{p_1}\right) \text{ for } 0 \le \mu(g) < \frac{1}{2},$$

$$\mu'(g) = \frac{1}{2}\left(1 + \left\{\frac{\sin\theta_2(2\mu(g)-1)}{\sin\theta_2}\right\}^{p_2}\right) \text{ for } \frac{1}{2} \le \mu(g) < 1$$

where,

$p1$	= exponential power used for black region,
$p2$	= exponential power used for white region,
θ_1	= angular equivalence for black region,
θ_2	= angular equivalence for white region,
$\mu(g)$	= fuzzy membership of the enhanced gray level image g.

This operator is capable of representing a wide range of transformations. It is a linear transformation when p_1 and p_2 are 1 and θ_1 and θ_2 are 0 and it approaches a step function as p_1 and p_2 approach 0.

Now that we have identified these three phases, fuzzy image contrast enhancement precedes as follows:

1. Scan through the image to determine l_0 and h_0, the minimum and maximum gray level of the original image.
2. Choose a value for m, the subjective threshold of the original image, and create a fuzzy set, G, using the fuzzification operator. This parameter should be a value near a cluster of gray-scale values on which the user is intrested in performing contrast enhancement.
3. After picking the values for p_1, p_2, θ_1 and θ_2, apply the fuzzy contrast enhancement operator to form a new fuzzy set, G'.
4. Create an enhanced image by choosing an appropriate value for n, and applying the defuzzification operator.

Genetic Algorithms

In several papers[1-8], the authors used a measure of image quality based on the degree of membership each pixel has in the category "black", and conversely, "white", to quantify the performance of their particular approach to contrast enhancement. Typically, the authors state that the lower the measures of fuzziness or image entropy, the better the image. Taking this assumption to the extreme, I've implemented a genetic algorithm to manipulate the various free parameters in De and Chatterji's [2] contrast enhancement operator so as to obtain the minimum measure of fuzziness. Before I describe my results in this area, an overview of the genetic algorithm I used to obtain these results is presented as follows:

Introduction to Genetic Algorithms

Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics. Survival of the fittest is combined with structured stochastic information exchange to

produce a simple but surprisingly powerful paradigm. The basic operations of a genetic algorithm are described by the following flow chart.

```
Generate an initial population, G(0)
Evaluate G(0)
t = 0
repeat
    t = t + 1
    generate G(t) using G(t-1)
    evaluate G(t)
until the solution is found
```
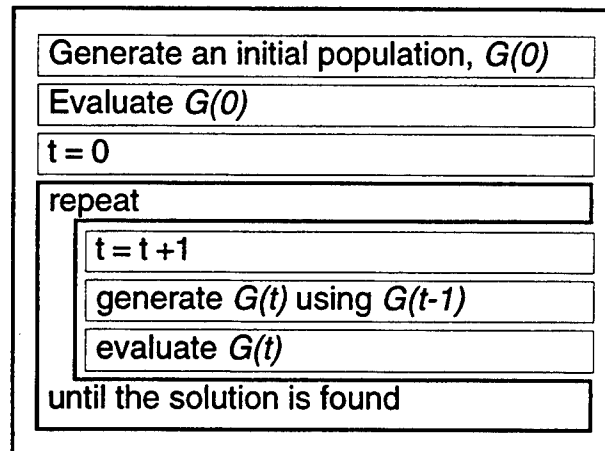
Historical Background

John Holland described the first genetic algorithm in his seminal work <u>Adaptation in Artificial and Natural Systems</u> published in 1975 [30], in the same year, Kenneth DeJong [33] provided careful empirical studies of GAs in the area of function optimization. He analyzed their performance over a wide variety of parameter settings using a test suite of 5 representative functions. These two works have provided the foundation for genetic algorithms and surprisingly few deviations from this introductory research have been explored since.

The surprising performance of genetic algorithms is closely related to results of statistical decision theory. More specifically k-armed bandits.

Statistical Decision Theory and K-Armed Bandits

A k-armed bandit is an extended version of a 2-armed bandit. Stated simply; you find yourself in front of a 2-armed slot machine with a pocketful of quarters. One side of the machine yields a better payoff than the other, and you want to play to maximize your profits. If the side with the higher average payoff is known, the problem is trivial, play all the quarters on the arm with the highest average. Otherwise, a strategy is needed to maximize the expected profits, but at the same time reduce the uncertainty about whether or not the arm being played is the correct one. Each quarter yields more information about the performance of each side; the goal is to exploit this information in the most quick and efficient manner possible.

Holland [30] has shown that the best strategy is to allocate an exponentially increasing number of trails (quarters) to the side with *observed* best payoff (after a pocketful of trials). Notice that this strategy is unrealizable since it requires knowledge of the outcome of a pocketful of trials before playing any of them; but it is useful, as it defines an upper-bound on the expected profits (performance) over a pocketful of quarters (trials).

This is an example of the trade-off between exploiting known information vs. exploring for new information. This strategy plays a key role in Selection component of genetic algorithms which will be discussed below.

Genetic Algorithm Components

Genetic algorithms are implemented as a randomly initialized population of individuals (*chromosomes*) upon which a sequence of directed, but random, combinations of simple operations

are applied. The subtle, yet complex, interaction among the operators produce a strategy very similar to the one described above.

Representation

The first step in using genetic algorithms is to define the meaning of the chromosome. Each chromosome is an encoded potential solution to a given problem. The normal representation scheme used to encode the chromosomes is a string over an alphabet (*allele*) of {0, 1}, where the position (*locus*) of each character encodes different partitions of the search space. Only one *allele* can be expressed at each position (*gene*) at a time. Some common examples are binary (base 2) and gray encoding. Another representation allows each *allele* to take values in the interval [0, 1] and a chromosome made of these types of alleles is called a floating point chromosome.
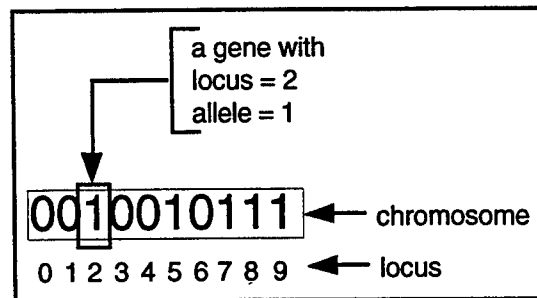
Fitness Function

After the representation scheme has been specified, a method must be provided for chromosomes to be compared with each other. This is the role of the *fitness function*. It is the definition of the problem to be solved and it provides a measure of "goodness" for each chromosome in the population. The fitness function may be discontinuous and highly nonlinear, but it must be non-negative and it must provide a totally ordered relation amongst the chromosomes.

Scaling

Typically, as evolution progresses in a GA, the population members become very similar and subsequently the variance of fitness values become very small. This is undesirable because "significant" differences (as measured through the fitness function) must be discernible amongst the population members in order for a GA to continue producing better solutions. Consequently, there may be various scaling methods applied to the basic fitness function to address this problem as well as the non-negativity constraint specified above.

Here is one approach for scaling the fitness function:

Power Law Scaling

The chromosomes' fitness is scaled as,

$$f'(x_i) = f^k(x_i), \qquad i = 1, 2, ..., N$$

where

N = number of chromosomes in the population.

x_i = the decoded value of chromosome i.

k = user defined positive constant near 1 (i.e. 1.005).

The parameter k scales the fitness function and it is usually problem dependent. This scaling method has been used successfully by Michalewicz [35] when the variable k is allowed to vary during the course of the genetic search. The exact nature of how k varies is controlled by a *scaling schedule* (very similar in concept to the cooling schedule in simulated annealing). The definition of the scaling schedule needs to be determined in advance, but how are we to know a good from a bad

schedule given a particular problem? The next section presents Michalewicz's [35] answer to this problem.

Non-Uniform Fitness Scaling

Ideally the scaling mechanism should adapt itself to the problem being optimized. Highly non-linear, discontinuous, multi-modal problems should place less emphasis on a chromosome's fitness in order to generate a more random search, while smooth, "well behaved" problems' should place more emphasis on chromosomes fitness in order to emphasize any differences in the population. Using this heuristic, we hope to provide a more intelligent sampling mechanism.

The question still remains of how to measure the behavior of a given fitness function. Michalewicz[35] proposes using a measure of the average function behavior with respect to its mean, or the *span* of a fitness function.

$$span = \frac{\sigma}{\mu}$$ and using the MVUEs $\hat{\sigma} = S$ and $\hat{\mu} = \overline{X}$ the estimated span is:

$$span_e = \frac{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(f(x_i)-\bar{f}\right)}}{\bar{f}}$$

where,

N = number of chromosomes in the population.

x_i = the decoded value of chromosome i.

$\bar{f}$ = population's mean fitness.

Michalewicz [35] has determined experimentally that fitness functions which have spans ~ 0.1 lend themselves well to genetic algorithms. These functions benefit most from a balanced trade-off between space exploration and speed of convergence. Functions with higher spans exhibit increasing amounts on non-smoothness and are consequently more difficult to optimize, while functions with low spans are linear in nature and are a very easy to optimize.

Finally we are ready to define the scaling schedule and we will do so using two dynamic parameters: *span* and *population age t*. The span usually needs to be calculated once after population initialization, while the variable t ranges from the value one to the maximum number of generations specified by the user. Michalewicz [35] defines the following Non-Uniform Scaling function which is a form of Power Law Scaling.

$$f'(x_i) = f^k(x_i) \text{ and } k = \left(\frac{0.1}{span_e}\right)^{p_1} \cdot \left(\tan\left(\frac{t}{T+1}\cdot\frac{\pi}{2}\right)\right)^{p_2\cdot(span_e/0.1)^{\alpha}}$$

where

p_1 = constant which determines the influence of the *span* on the magnitude of k.

p_2 = constant which determines the speed at which k changes.

α = another constant which determines the speed at which k changes.

T = maximum number of generations.

The following three figures illustrate the effects of various spans on the scaling schedule. The parameters α, p_1, and p_2 are set to 0.1, 0.05, and 0.1 respectively.
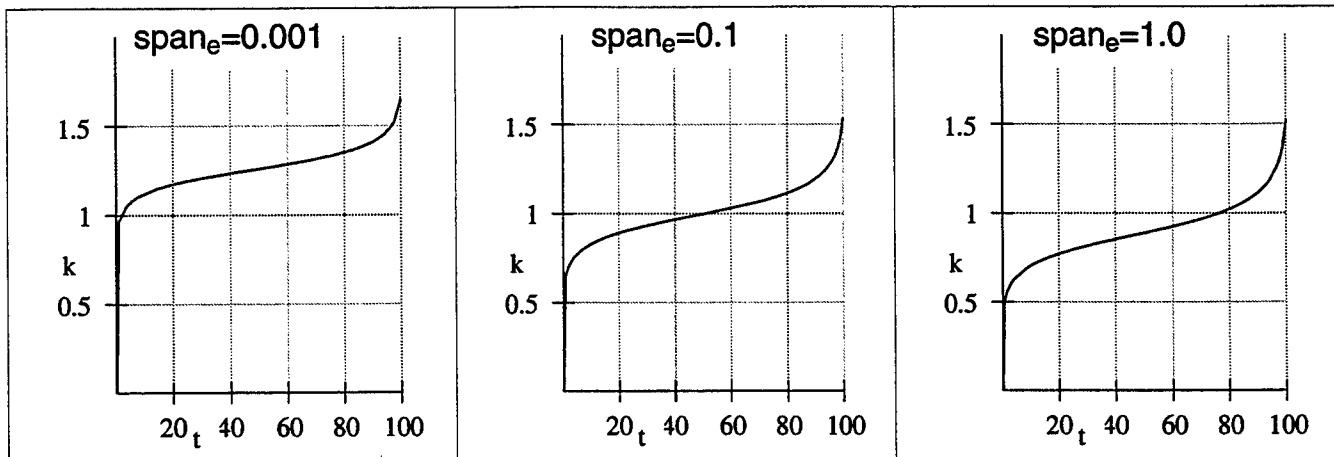
You can see that when the span is low, (~0.001), the scaling function produces values of k greater than 1.0 through out the entire schedule. This is in order to emphasize any differences in what may be a very homogenous search space. As the span increases (~0.1), the scaling function produces a more random selection mechanism in the first half of the genetic algorithm, and then increases selection pressure in the second half of the run. This behavior is based on the belief that by the midpoint, the GA has converged to near optimal points and more selection pressure is required to find the global optimum. When the span is high (~1.0) it indicates that the search space is highly non-linear and perhaps discontinuous. The consistently low values of k during the scaling schedule, produce a randomizing effect which is required to find the optimal point in a very "hilly" search space.

Selection

Selection is the method which comes very close to realizing the k-armed bandit strategy discussed above. As the name implies, it is the method used to <u>select</u> the chromosomes in the current generation for inclusion in the next. It gives an exponentially increasing number of trials to the observed best (as seen through the fitness function) members of the population and an exponentially decreasing number of trials to the observed worst.

There are a great variety of ways to realize this strategy, One of the most common is listed below. (Note that N is the number of chromosomes in the current population.)

Stochastic Universal Sampling
In 1987 Baker [37] presented a comprehensive theoretical study of various selection mechanisms using some well defined measures. He also presented a new an improved method which uses a single spin of a roulette wheel to select N parents. The wheel, which is constructed with slots sized in proportion to the chromosome's contribution to the total fitness, has N equally spaced pointers around the circumference. A single wheel spin can then select N parents for inclusion into the next generation. The details are as follows:

- Calculate the total fitness of the population
$$F = \sum_{i=1}^{N} fitness(c_i), \qquad \text{where } c_i \text{ is chromosome } i$$
- Calculate a *probability of selection* for each chromosome in the population.
$$p_i = \frac{fitness(c_i)}{F}$$
- Calculate a *cumulative probability* for each chromosome in the population.
$$q_i = \sum_{j=1}^{i} p_j$$

12-11

- Generate one random number $x_1 \in \left(0, \dfrac{1}{N}\right)$. For each number x_j, where

$x_{j+1} = x_j + \dfrac{1}{N}$, $j = 1 \ldots N$, <u>select</u> the chromosome i whose cumulative probability is the greatest lower bound of x_j (i.e. $q_{i-1} < x_j < q_i$).

Reproduction Operators

So far we have only discussed methods of selecting members of a current population for inclusion in the next. While these methods faithfully implement the k-armed bandit strategy, they do not introduce new potential solutions into consideration. Selection is strictly concerned with exploiting known information. This section, and the next, will discuss methods of introducing new and hopefully better candidates into our population of potential solutions.

<u>Crossover</u>

The classic *crossover* operator is a simple one point representation of the crossover in biological genetics. Just as artificial neural networks exhibit the "important" properties of neuron interactions, the classic crossover captures the "essence" of the biological crossover, that of creating new population members from a combination of existing ones. The following is a description of the crossover operator used in genetic algorithms.

The crossover operation is governed by a *Probability of Crossover* P_c

For each chromosome of the newly <u>selected</u> population:
- Generate a random number r in [0, 1]
- If $r < P_c$ then mark this chromosome as available for crossover.

After all of the population members have been checked for crossover, we have a small pool of potential parents.

Do the following until the pool is empty or only one parent remains.

- Randomly pick two parents from the pool to be mated.
- Generate a random number r in {2, ..., l-1}, where l is the number of genes in the chromosome.
- Create two new chromosomes by taking the first r genes from parent #1 and the remaining from parent #2 and (vice versa). These two new chromosomes replace their parents in the current population.
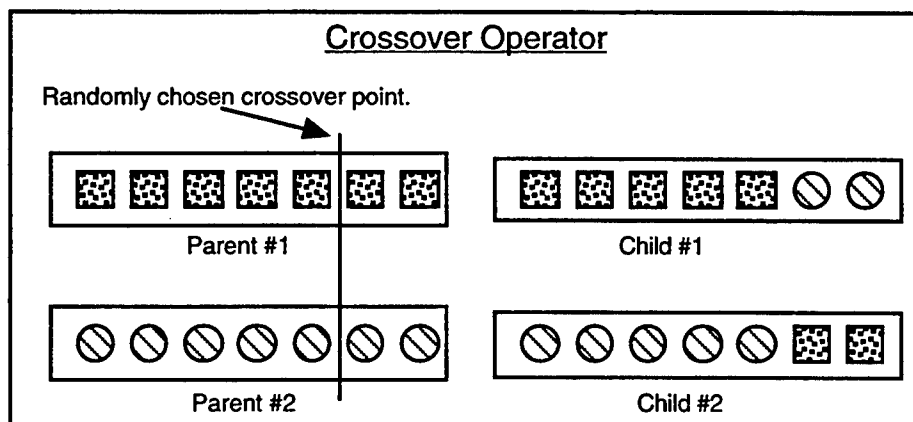
<u>Mutation</u>

Selection and crossover are the predominate operators in genetic algorithms, but they are not capable of introducing new alleles into the population. Recall that an allele is a specific value of a gene at a specific position. For example, if at the initialization of the GA, the allele 1 at gene position 4 is not expressed in any chromosome of the population, then under crossover and selection, it *never* will be. The *mutation* operator was designed to prevent such a situation. Under a small but positive probability, each gene of each chromosome has a chance of being changed to a random member from the defining alphabet at that position. The following is description of the mutation operator in genetic algorithms.

The mutation operation is governed by a *Probability of Mutation* P_m

For <u>each gene</u> of <u>each chromosome</u> in the population:
- Generate a random number r in [0, 1]
- If $r < P_m$ then generate another random number t in {1, 2, ..., n}, where n is cardinality of the defining alphabet at that position.
- Change the allele at the current position to the character of the alphabet corresponding to the random number t.
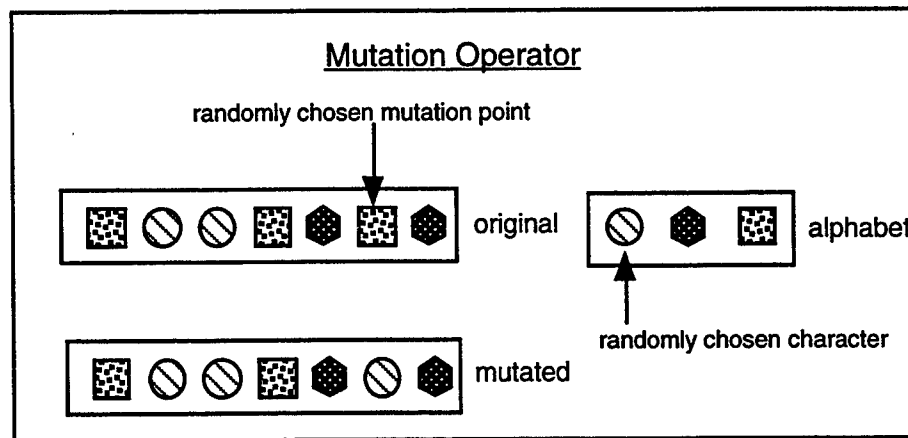
<u>Uniform Mutation</u>
Given a chromosome $\bar{x} = (x_1, x_2, ..., x_n)$ of n floating point numbers, a *uniform mutation* is defined as:

- Choose a random integer i from {1, n}
- Replace the i-*th* component of $\bar{x}$ with a uniformly distributed random number from the domain of the i-*th* component of $\bar{x}$. i.e. $\bar{x}_{new} = (x_1, x_2, ..., x_i', ..., x_n)$

The uniform mutation operator is most similar to that of the classic mutation operator used with binary encodings.

<u>Non-Uniform Mutation</u>
Given a chromosome $\bar{x} = (x_1, x_2, ..., x_n)$ of n floating point numbers, a *non-uniform mutation* is defined as:

- Choose a random integer i from {1, n}.
- Choose a random number d from {0, 1}.
- Choose a random number r from [0, 1].

- Replace the *i-th* component of $\bar{x}$ with $x_i' = \begin{cases} x_i + \Delta(t, u_i - x_i) & \text{if } d = 0 \\ x_i - \Delta(t, x_i - l_i) & \text{if } d = 1 \end{cases}$, where $[l_i, u_i]$ is the

  domain of the *i-th* component of $\bar{x}$ and $\Delta(t, y) = y \cdot \left(1 - r^{\left(1 - \frac{t}{T}\right)^b}\right)$ (b is a system parameter

  determining the degree of non-uniformity, t is the current generation, and T is the maximum generation). i.e. $\bar{x}_{new} = (x_1, x_2, ..., x_i', ..., x_n)$

The operator has the property of producing near uniform random changes early on in the genetic search and very small random changes late in the genetic search. This is good for local tuning of near optimal solutions. The exact properties of this tuning, of course, depends on the Δ function. The following figures display this function's behavior at three different times with $b = 2.0$ and $y = 1.0$. This should provide a good feel for how this operator works.
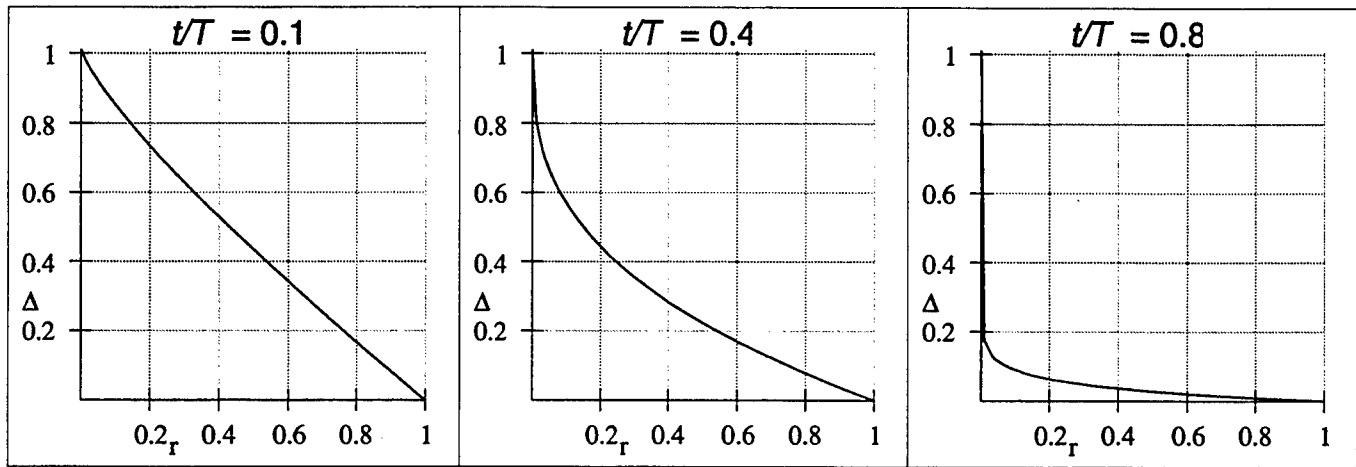
Simple Crossover
Given two chromosomes $\bar{x} = (x_1, x_2, ..., x_n)$ and $\bar{y} = (y_1, y_2, ..., y_n)$ each of n floating point numbers, a *simple crossover* is defined as:

- Choose a random integer i from $\{1, n\}$.
- Create one child by taking first i components from $\bar{x}$. and the last $n-i+1$ components from $\bar{y}$.
  i.e. $\bar{x}_{new} = (x_1, x_1, ..., x_i, y_{i+1}, ..., y_n)$
- Create another child by taking first i components from $\bar{y}$. and the last $n-i+1$ components from $\bar{x}$. i.e. $\bar{y}_{new} = (y_1, y_1, ..., y_i, x_{i+1}, ..., x_n)$

Arithmetic Static Crossover
Given two chromosomes $\bar{x} = (x_1, x_2, ..., x_n)$ and $\bar{y} = (y_1, y_2, ..., y_n)$ each of n floating point numbers, a *arithmetical static crossover* is defined as the linear combination of $\bar{x}$ and $\bar{y}$.

- $\bar{x}_{new} = (a \cdot x_1 + (1-a) \cdot y_1, a \cdot x_2 + (1-a) \cdot y_2, ..., a \cdot x_n + (1-a) \cdot y_n)$
- $\bar{y}_{new} = (a \cdot y_1 + (1-a) \cdot x_1, a \cdot y_2 + (1-a) \cdot x_2, ..., a \cdot y_n + (1-a) \cdot x_n)$

Where is a constant system parameter.

Arithmetic Random Crossover

This operator is defined exactly as Arithmetic Static Crossover except the parameter a is a random number from a uniform distribution over [0,1] for each gene in the chromosome.

Dynamic Operator Adaptation

Discovering the optimal parameter values of a GA is not a trivial task. The interdependencies of each part of the GA make it almost impossible to determine optimal values for any given implementation. Also, as noted in Davis [32], the optimal parameter settings often change during the evolution of a GA. How are we to get a grasp on all of this complexity in order to make a GA solve our problems of interest?

The first option is to carry out hand optimization which combines human intuition and extensive amounts of computer time. This is the approach DeJong [33] took in his doctoral dissertation.

The second option, as pioneered by Grefenstette [29], is to use a higher level GA to optimize the parameter settings. But the question still remains of the parameter settings for the higher level GA.

The third option is by exhaustive search. Michalewicz [35] states that Scaffer, Eshelman, Caruana, and Das consumed more than 12 months of CPU time in finding optimal static parameter settings for the DeJong test suite and a few problems of their own. Their final conclusion was that optimal setting vary from problem to problem, but there exist robust settings which perform reasonably well across a wide range of problems.

A fourth option, and the one I've taken in my summer research at Rome Labs, is to allow the parameter settings to evolve along with the population of potential solutions. This approach was pioneered by Davis [32] and consists of measuring reproductive operator fitness, adjusting operator probabilities to reflect this measure, and finding good initial values for operator probabilities. Each of these three parts is described below:

Computing Operator Performance
1. Whenever a new image is created, record the parents and the operator which created it.
2. If the new image is better than the best image in the population, give it an amount of credit equal to the amount that its image quality exceeds the population's image quality.
3. Whenever a new image is given credit for a better evaluation, pass a portion of that credit back to the parents of the new image. The number of generations to pass back and the proportion of credit for each generation are parameters of this algorithm.
4. To compute the performance of an operator for a given period, sum the credit of each image the operator produced in that period and divide it by the number of images produced. The type of period and its length are a parameters of this algorithm.

Adapting Operator Probabilities
Let $x<1$ be the amount of operator probability that we wish to adapt.
1. If all operator performance measures are zero, then do nothing.
2. Otherwise, multiply the list of operator probabilities by a constant factor so that they total $1-x$. Call this list the base probability list.
3. Form a list of operator performance measures over the recent period.
4. Multiply the list of operator performance measures by a constant factor so they total x.
5. Add the operator's performance to its base probability to form the new operator probability.

Initializing Operator Probabilities
1. Set all operator probabilities to be identical, unless better values are known.
2. Initialize the population and remember each image and its associated quality.

3. Run the genetic algorithm on the initial population to create k new images, where k is the length of the period between adaptations.
4. Adapt and then remember the operator probabilities.
5. Repeat steps 3 and 4 once.
6. Average the operator probabilities derived in steps 2-5 and set the operator probabilities to this average.
7. Using the new operator probabilities, repeat steps 2-6 until the difference from one loop iteration to the next, of operator probabilities derived step 6, is negligible.

Davis [32] has had good luck with adapting 15% of the operator probabilities at periods of 50 individuals

Initialization
There are many ways of creating the initial population of images. The approach I've taken is to randomly choose values of each enhancement parameter and then run Davis's [32] dynamic operator probability initialization algorithm.

Parameters
There are some "user provided" parameters which must be defined before a GA can operate such as:

- N = population size
- Stopping criteria

The choices for these parameters will greatly effect the performance of the GA. A lot of empirical studies have been done with very little conclusive results, but as a general rule, the following should provide reasonable performance:

- N = [50, 500]
- Execute between [50,5000] generations

Genetic Algorithm Optimized Fuzzy Contrast Enhancement
The first task that needs to be addressed before a GA can be applied to optimizing the free parameters of De and Chatterji's [2] fuzzy contrast enhancement operator is that of an appropriate fitness function. As I've mentioned before, several authors [1-8] have justified their unique approach to fuzzy contrast enhancement by using various measures of fuzziness to compare their results to other results in the literature. The typical conclusion is as follows, "as can be seen in table X, our approach consistently produces images of lower uncertainty, and thus better image quality." Taking this observation to the extreme, we use a genetic algorithm to minimize a measure of fuzziness to the greatest possible extent hoping that this will lead to the best image possible.

Because of the intense computational complexity involved in a evaluating this particular fitness function, (each pixel of the image represented by each chromosome must be examined at every generation), I've implemented the GA on the DEC Alpha computer and applied it to the task of optimizing a 132 x 88 (11,616 pixels) image of an aircraft parked on a runway.

I used a population of 200 with a termination requirement of 50 generations. I fixed θ_1 and θ_2 at 0.5 and optimized for m, p_1, and p_2, as they seemed to be the most influential parameters. These parameters were represented by floating point chromosomes and I used Davis's [32] algorithm to dynamically tune the floating point operator probabilities during the course of a run. Bakers[37] Stochastic Universal Sampling technique was used along with Michalewicz's[35] non-linear fitness

scaling algorithm. Each run of the GA took approximately 15 minutes and some typical results are presented below:
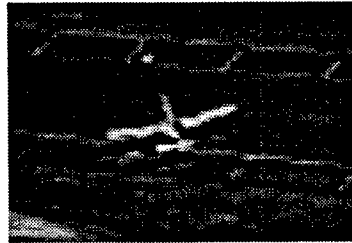
Original Image
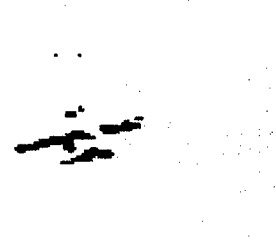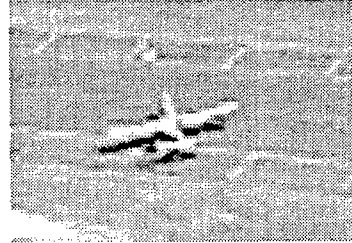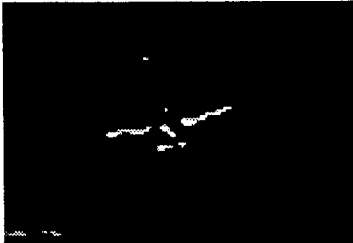
Enhanced images with $(m, p1, p2)$ set to (56360, 0.05, 0.9), (3278, 0.007, 0.43), and (16384, 0.97, 0.01)

As you can see from the images, the naive approach of minimizing the image fuzziness leads to black and white pictures. A movements reflection will indicate that this is exactly as expected. If the abstract concepts we are attempting to classify are black and white, then a pure black and white image is one of minimal fuzziness. This result may be of limited usefulness to the image analyst. A better measure of image quality is necessary for this approach to produce more meaningful results. Due to time and resource constraints this avenue of investigation was not pursued further, however it may prove to be a very useful research topic as explained below.

Overview of Software

In order to facilitate experimentation with De and Chatterji's [2] fuzzy contrast enhancement operator, Macintosh and Sparc Station software has been created which allows an intuitive interface to the six free parameters of this operator and provides immediate visual feedback regarding the output image given particular settings of the input parameters. This allows an image analyst to judge the resultant image quality in near real-time. After an the parameters have been set to produce an subjectively pleasing image, a mechanism has been implemented which allows the image analyst to save the image to a industry standard TIFF formatted file.

The software also supports 24 bit RGB color images. The software will first convert the RGB image into a YIQ image, apply the contrast enhancement operator to the Y, intensity, component, and then convert the enhanced image back to RGB for display. As mentioned before, any resultant output image can be saved in a TIFF formatted file.

The near real-time performance of this software is due to direct manipulation of the underlying color palette hardware. In most moderately priced computer visual systems there exits a color palette, which is a mapping from 256 possible pixel values to 2^{16^3} distinct colors, that determines the colors displayed on the screen. This mapping is implemented in computer hardware and is capable of refreshing the screen some where between 60 and 75 times per second. The color palette display system operates as follows:

1. The software, under programmer control, writes a pixel value between 0 and 255 into video ram.
2. The computer display subsystem, under hardware control, reads this number and uses it as an index into a color palette.

3. The color palette contains three numbers between 0 and 65535 which represent the amount of Red, Green, and Blue that should be displayed on the screen for this particular pixel value.
4. The computer display subsystem takes these three numbers and feeds them through a D/A (Digital to Analog converter) to produce RGB output voltages which are then feed to a monitor and subsequently displayed on the screen.

If you've followed me so far, you now know that there can be at most 256 different colors displayed on the screen at the same time. Realizing this, the fuzzy contrast enhancement software will operate directly on these 256 colors in the color palette, producing new color palette entries, which then are displayed on the screen by the computer hardware. So, the initial task of manipulating *width * height* pixel values, has been replaced by one which manipulates color palette entries and thus dramatically reduces the computational complexity involved in the problem and allows near real-time performance.

The Genetic Algorithm software has been implemented in ANSI-C and runs on 175Mhz DEC Alpha work station as compiled using the GNU gcc tool set. The Macintosh software has been implemented in C++ under the Symmantic Think Class library, and the Sparc Station software has been implemented in ANSI-C under the Motif Widget Library. This software is rated to be of "research" quality. That is, it was designed as a platform for facilitating experimentation with this contrast enhancement operator rather than as a robust, multi-featured, production product.

To give the reader some feel for the computational complexity involved in the genetic algorithm, which operates on pixel values and not palette entries (see Future Research), I will relate the run times of the genetic algorithm on a DEC Alpha (175Mhz 21064a), a 110Mhz Power Mac, and a Macintosh Quadra (33Mhz 68040).

	175Mhz 21064a	110Mhz Power PC	33Mhz 68040
Running Time	5 minutes	1 hour 15 minutes	7 hours 30 minutes
Normalized Time	1X	15X	90X

This software has been demonstrated to the Central Imagery Office with positive, encouraging feedback received for research in this area.

Conclusions and Future Work

Directly applying genetic optimization techniques to the minimization of fuzziness in the image contrast enhancement operator has lead to marginal success. A better measure of image quality is needed in order for the genetic algorithm to produce quality output images. One approach to this end might be to add constraints to the problem similar to those imposed in Maximum Entropy Reconstruction mentioned earlier in this paper. Another approach, which was developed by Fang and Cheng[1], is to use Bayesian statistical techniques to optimize the selection of the parameter m. This idea could be extended to selection of all six free parameters. An alternative to this would be to combine several different image quality operators together using a non-additive measure to produce better quality genetic algorithm solutions. This concept is best described in the following example of a Television Rating Problem.

Consider the task of rating the overall quality of a television set given individual ratings of picture and sound. An additive approach would be a weighted average:

$$q = \alpha p + (1 - \alpha)s \text{ with } q, p, s, \alpha \in [0,1]$$

where q is overall quality, p is quality of picture, s is quality of sound, and a is the weight.

It is possible to show that this model can not represent the intuitive requirement that a television with perfect picture and no sound or one with perfect sound and no picture be rated lower in quality than one

with mediocre picture and sound. In other words, in terms of overall usefulness, the importance of having *both* picture and sound is higher than that of having picture or sound individually. Fuzzy Measure theory [28] can capture this reasoning by assigning an importance measure as follows:

Note:

$$\mu(\{p,s\}) = 1.0, \ \mu(\{p\}) = 0.4, \ \mu(\{s\}) = 0.3, \text{ where } \mu \text{ is a measure of importance.}$$

$$\mu(\{p,s\}) > \mu(\{p\}) + \mu(\{s\}) \text{ and } \{p,s\} = \{p\} \cup \{s\}$$

This is a super-additive model since the measure of the whole is greater than the sum of the measures of its parts.

The individual quality ratings can expressed as a function from the set of measurable quantities (sound, picture) to values in the closed interval, zero-one.

$$f: X \rightarrow [0,1] \text{ where } X = \{p,s\}$$

A model of overall quality which captures the non-additive reasoning described above can be expressed by the fuzzy integral:

$$q = \int_X f(x) d\mu$$

A non-additive combination of image quality measures would be able to capture the importance of various groupings of the individual measures. This approach would provide a means for "information fusion" of more expressive power than the classical additive approaches and may lead to better output images.

On another note, it seems as if it would be possible to extend the idea of direct color palette manipulation to the genetic algorithm by the use of image histograms. If this were successful, then there would be a tremendous improvement in the computational complexity involved in the problem, so much so in fact, that it would become feasible to apply this technique to "real-world" images which have typical of dimensions 1024x1024 or 1,048,576 pixel values.

Bibliography

1. Fang, N., and M. Cheng. May 1993. An automatic crossover point selection technique for image enhancement using fuzy sets. *Pattern Recognition Letters* 14 (5): 397-406.

2. De, T. K., and B. N. Chatterji. 1988. An approach to a generalised technique for image contrast enhancement using the concept of fuzzy set. *Fuzzy Sets and Systems* 25: 145-158.

3. Huang, L., and M. J. Wang. 1995. Image thresholding by minimizing the measures of fuzziness. *Pattern Recognition* 28 (1): 41-51.

4. Li, H., and H. S. Yang. October 1989. Fast and reliable image enhancement using fuzzy relaxation technique. *IEEE Transactions on systems, man, and cybernetics* 19 (5): 1276-1281.

5. Li, X., Z. Zhao, and H. D. Cheng. 1995. Fuzzy entropy threshold approach to breast cancer detection. *Information Sciences* 4: 49-56.

6. Murthy, C. A., and S. K. Pal. 1990. Fuzzy thresholding: mathematical framework, bound functins and weighted moving average technique. *Pattern Recognition Letters* 11: 197-206.

7. Kundu, M. K., and S. K. Pal. 1990. Automatic selection of object enhancement operator with quantitative justification based on fuzzy set theoretic measures. *Pattern Recognition Letters* 11: 811-829.

8. Pal, S. K., and R. A. King. July 1981. Image enhancement using smoothing with fuzzy sets. *IEEE Transactions on Systems, Man, and Cybernetics* 11 (7): 494-501.

9. Pal, S. K. 1989. Fuzzy skeletonization of an image. *Pattern Recognition Letters* 10: 17-23.

10. Dave, R. N., and T. Fu. 1994. Robust shape detection using fuzzy clustering: practical applications. *Fuzzy Sets and Systems* 65: 161-185.

11. Pal, S. K., and S. Mitra. 1990. Fuzzy dynamic clustering algorithm. *Pattern Recognition Letters* 11: 525-535.

12. Hirota, K., and W. Pedrycz. 1995. D-fuzzy clustering. *Pattern Recognition Letters* 16: 193-200.

13. Krishnapuram, R., H. Frigui, and O. Nasraoui. 1993. The fuzzy c quadric shell clustering algorithm and the detection of second-degree curves. *Pattern Recognition Letters* 14 (7): 545-552.

14. Trivedi, M. M., and J. C. Bezdek. August 1986. Low-level segmentation of aerial images with fuzzy clustering. *IEEE Transactions on Systems, Man, and Cybernetics* 16 (4): 589-598.

15. Schroeter, P., and J. Bigun. 1995. Hierarchical image segmentation by multi-dimensional clustering and orientation-adaptive boundary refinement. *Pattern Recognition* 28 (5): 695-709.

16. Bhandarkar, S. M. May 1994. A fuzzy probabilistic model for the generalized hough transform. *IEEE Transactions on Systems, Man, and Cybernetics* 24 (5): 745-759.

17. Han, J. H., L. T. Koczy, and T. Poston. 1994. Fuzzy hough transform. *Pattern Recognition Letters* 15: 649-658.

18. Tahani, H., and J. M. Keller. May 1990. Information fusion in computer vision using the fuzzy integral. *IEEE Transactions on Systems, Man, and Cybernetics* 20 (3): 733-742.

19. Pal, N. R., and S. K. Pal. September 1991. Entropy: a new definition and its applications. *IEEE Transactions on Systems, Man, and Cybernetics* 21 (5): 1260-1270.

20. Higashi, M., and G. J. Klir. 1982. On measures of fuzziness and fuzzy complements. *International Journal of General Systems* 8: 169-180.

21. Jain, A. K. 1989. *Fundamentals of digital image processing*. Englewood Cliffs, New Jersey: Prentice Hall.

22. Russ, J. C. 1995. *The image processing handbood, 2nd edition*. Boca Raton, Ann Arbor, London, Tokyo: CRC Press.

23. Rosenfeld, A., and A. C. Kak. 1982. *Digital picture processing, second edition, volume 1*. New York, London, Paris, San Diego, San Francisco, Sao Paulo, Sydney, Tokyo, Toronto: Academic Press.

24. ---. 1982. *Digitial picture processing, second edition, volume 2*. New York, London, Paris, San Diego, San Francisco, Sao Paulo, Sydney, Tokyo, Toronto: Academic Press.

25. Bezdek, J. C., and S. K. Pal. 1992. *Fuzzy models for pattern recognition*. New York: IEEE Press.

26. Dubois, D., and H. Prade. 1988. *Possibility theory*. New York and London: Plenum Press.

27. Klir, G., and T. A. Folger. 1988. *Fuzzy sets, uncertainty, and information*. Englewood Cliffs, New Jersey: Prentice Hall.

28. Wang, Z., and G. J. Klir. 1992. *Fuzzy measure theory*. New York and London: Plenum Press.

29. Grefenstette, J. J. 1986. Optimization of control of parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics* 16 (1): 122-128.

30. Holland, J. H. 1975. *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.

31. Klir, G. J., and B. Yuan. 1995. *Fuzzy sets and fuzzy logic*. Upper Saddle River, New Jersey: Prentice Hall.

32. Davis, L. 1991. *Handbook of genetic algorithms*. New York, New York: Van Nostrand Reinhold.

33. De Jong, K. A. 1975. An analysis of the behavior of a class of genetic adaptive systems. Ph.D. diss., The University of Michigan, Michigan.

34. Goldberg, D. E. 1989. *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison Wesley.

35. Michalewicz, Z. 1992. *Genetic algorithms + data structures = evolutionary programs*. Berlin: Springer-Verlag.

36. Zadeh, L. A. August 1984. Making computers think like people. *IEEE Spectrum* 26-32.

37. Baker, J. E. 1987. Reducing bias and inefficiency in the selection algorithm. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms* 14-21.

38. Weirman, M. 1994. Possiblistic image processing. Ph.D. diss., SUNY-Binghamton, Binghamton, NY.

A HYBRID MM/GTD NUMERICAL TECHNIQUE FOR LOSSY DIELECTRIC ROUGH SURFACE SCATTERING CALCULATIONS

James Michael Sturm
Graduate Research Assistant
School of Electrical Engineering

Oklahoma State University
202 ES Stillwater, OK 74078

A HYBRID MM/GTD NUMERICAL TECHNIQUE FOR LOSSY DIELECTRIC ROUGH SURFACE SCATTERING CALCULATIONS

James C. West
Associate Professor

James Michael Sturm
Graduate Research Assistant

School of Electrical and Computer Engineering
Oklahoma State University

Abstract

A hybrid numerical technique combining the moment method and the geometrical theory of diffraction has been extended to allow the calculation of electromagnetic scatter from lossy dielectric surfaces. The hybrid technique eliminates the non-physical edge effects that are introduced in standard moment method implementations, thereby allowing the application at extreme grazing angles. The dielectric surface is represented using impedance boundary conditions. Sample calculations demonstrate the reduction in scattering from a rounded-apex wedge when the surface conductivity is reduced. The technique should allow more realistic calculation of the scattering from land and water surfaces than can be obtained using a perfectly conducting surface.

I. INTRODUCTION

Traditional implementations of the moment method for calculating the electromagnetic scattering from rough surfaces have been limited to application at moderate to large illumination grazing angles [1, 2]. Finite computer resources limit the length of the surface that can be numerically modeled, introducing non-physical edges in the scattering surface that can lead to unrealistic diffractive scattering. These "edge-effects" are often avoided using an illumination weighting function that reduces the incident field to negligible levels at the edges. Thorsos [3] showed that electromagnetically valid weighting functions can yield unrealistic surface illumination if the modeled surface is insufficiently large. Unfortunately, the required surface length increases dramatically with decreasing grazing angle (increasing incidence angle), limiting the smallest grazing angle at which the technique can be applied.

West [4] implemented a hybrid numerical technique combining the moment method (MM) and the geometrical theory of diffraction (GTD) that overcomes many of the limitations of the traditional moment method when calculating the scattering from perfectly conducting surfaces. In this approach the surface is extended to infinity, thereby eliminating the artificial edges. The technique has been used to investigate the effects of surface self-shadowing on the backscattering from perfectly conducting surfaces approximating rough ocean waves [4] and the effects of multi-path on backscattering from breaking ocean waves [5]. This hybrid technique has been enhanced to allow the application to lossy dielectric surfaces, thereby allowing accurate prediction of the scattering from rough land surfaces as well as sea surfaces. A detailed description of the enhanced technique is given below, as well as sample calculations of the scattering from lossy dielectric objects.

II. OVERVIEW OF TECHNIQUE

A. *Perfectly Conducting Surfaces*

Application of the hybrid MM/GTD numerical technique to scattering from perfectly conducting surface is described in [4]. A detailed review is given here. Adapted from the technique described by Burnside *et al.*[6], it is quite similar to the standard moment method in that scattering from the surface is found by first numerically solving an integro-differential equation to yield the surface current. In both methods, the unknown surface current is represented as a summation of known basis functions. The weighting coefficients associated with each basis function that give the "best" approximate solution are obtained using the moment method. The primary difference between the two techniques is that the hybrid approach uses *a priori* knowledge of the current obtained from GTD to define well behaved basis functions and additional source terms that allow the treatment of special infinitely long surfaces, thereby avoiding the artificial edge effects introduced in the standard MM. The surface current is then radiated to yield the scattered field.

For the one-dimensionally rough surfaces considered here, vertically polarized scattering is best described by the magnetic field integral equation (MFIE) [7]:

$$
\begin{aligned}
H^i(l) &= 0.5 J_s(l) + j\frac{\beta}{4} \int J_s(l')\,(\hat{n}' \cdot \rho')\, H_1^{(2)}(\beta|\rho - \rho'|)\, dl' \\
&= L_M[J_s(l)],
\end{aligned}
\tag{1}
$$

where l is the arc length along the scattering surface, $H^i(l)$ is the incident magnetic field at the scattering surface, $J_s(l)$ is the unknown surface current to be found, β is the free space wave number, ρ is the position vector of the observation point, ρ' is the position vector of the source point, $\hat{n}'$ is the normal unit vector at the source point, and $H_1^{(2)}$ is the first-order Hankel function of the second type. The integration is the principal value integral (avoiding the singularity where $l = l'$) over the entire surface. Horizontally polarized scattering is more
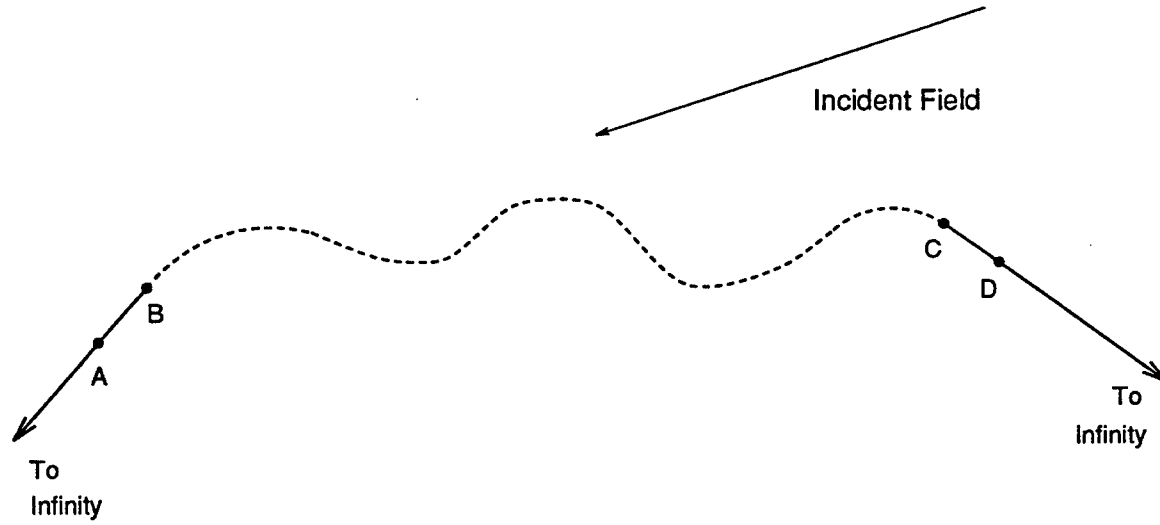
Figure 1: Arbitrary scattering surface.

easily treated by the electric field integral equation (EFIE):

$$
\begin{aligned}
E^i(l) &= \frac{\beta \eta_0}{4} \int J_s(l') H_0^{(2)}(\beta |\rho - \rho'|)\, dl' \\
&= L_E[J_s(l)],
\end{aligned}
\tag{2}
$$

where η_0 is the intrinsic wave impedance of free space. The MFIE and EFIE can be written in the single notation

$$
F^i(l) = L_X[J_s(l)],
\tag{3}
$$

where F is either E or H and X is either E or M. In the standard moment method, the infinite integrations in equations (1) and (2) are truncated to be over a finite surface arc length. The current on the modeled length L is then divided into a weighted summation of adjacent pulse basis functions, and the moment method is used to find the associated weighting coefficients. It is the truncation of the integrations that lead to the non-physical edge effects.

The hybrid technique is applied to one-dimensionally rough surfaces of the form shown in Figure 1. The dashed section of the surface represents the actual rough surface while the solid line represents infinitely long, planar extensions. The extensions are chosen such that all

points on the actual surface are shadowed from all points on the extension (except of course at the intersection points B and C). Because the surface is arbitrary, little is known initially about the current between points A and D. Thus, the current in this region is described using standard MM pulse basis functions with impulse testing functions (yielding point matching) centered on the basis functions.

Since the extensions are shadowed from the arbitrary surface points, the fields at the surface of the extensions can be entirely described as the sum of a field diffracted from point B or C plus the geometrical optical (GO) incident and reflected fields:

$$F^t = F^i + F^s = F^{GO} + F^d, \tag{4}$$

where F^t is the total field, F^i is the incident field, F^s is the scattered field, F^{GO} is the gemetrical optics incident and reflected fields, and F^d is the diffracted field. The current on the extension is obtained by applying the surface boundary conditions to equation (4), yielding the physical optics current associated with the GO fields plus an additional current component associated with the diffracted field (the "diffraction-field current"):

$$J_s = J_{PO} + J_d. \tag{5}$$

Since the extension is flat and perfectly conducting, the PO current is known exactly *a priori*. (Note that if the extension is shadowed from the incident field the PO current is simply zero). However, the diffracted field, and therefore the diffraction-field current, is not known initially and must be determined using the moment method. Since it extends to infinity, use of ordinary sub-domain MM basis functions to describe this current would lead to an infinite order system of linear equations that cannot be solved. Instead it is recognized that at distances far enough away from the diffraction point the diffracted field is ray optical. Thus,

E, H   E, H

$$J_s = n \times H$$
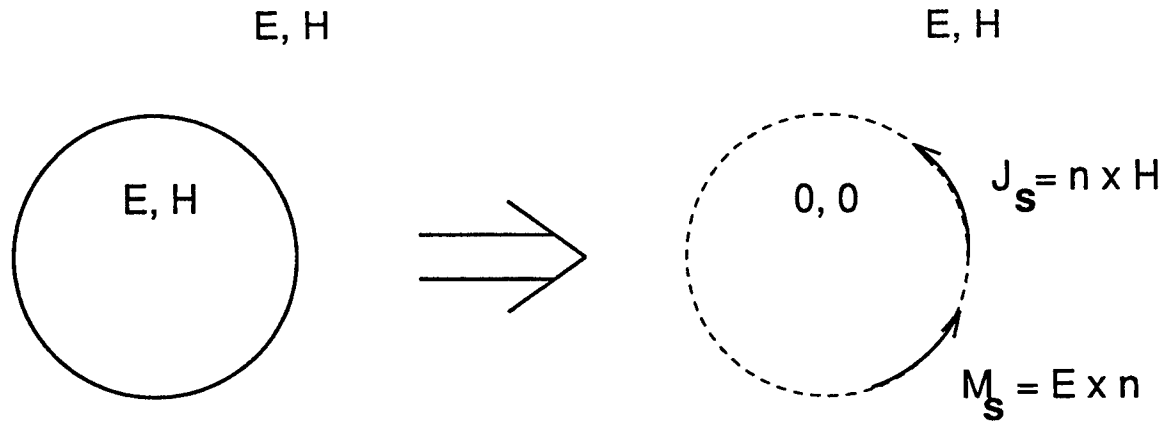
$$0, 0$$

$$M_s = E \times n$$

Figure 3: Equivalent problem to be solved with lossy dielectric scatterer.

completing the MM solution of the current. The far field scatter is then determined from

$$F^s = -L_X[J_{MM} + J_D + J_{PO}]\big|_{r \to \infty}. \tag{14}$$

B. *Lossy Dielectric Surfaces*

When the scattering surface is perfectly conducting a true surface current exists. Thus, the moment method solves the physical scattering problem directly. When the surface is not perfectly conducting a surface current cannot be supported; the field penetrates the surface and a volume current density exists. The moment method is not well suited for direct application to volume current problems. Instead, the equivalence principle [7] is applied as shown in Figure 3, yielding both electric (J) and magnetic (M) surface current densities that radiate the desired scattered field. Although the equivalent problem includes only surface currents, the moment method still cannot be applied directly since the unknown electric and magnetic currents are co-located. Instead, the magnetic current is expressed in terms of the electric current using impedance boundary conditions [8]. Assuming that the conditions

$$|N| \gg 1, \qquad |\text{Im}(N)k\rho_l| \gg 1 \tag{15}$$

where N is the complex refractive index of the scattering medium and ρ_l is the radius of curvature of the surface, are met everywhere on the surface, the field penetrating into the surface propagates as a plane wave in the negative surface normal direction. The two surface current components can then be related by [9]

$$\mathbf{M} = -Z_s \hat{n} \times \mathbf{J}, \tag{16}$$

where Z_s is the intrinsic wave impedance of the lossy dielectric.

Applying duality to equations (1) and (2) to determine the near-field radiation of the magnetic current density and using equation (16), it is straightforward to show that the appropriate two-dimensional MFIE for determining vertically polarized scattering from a lossy dielectric scatterer is [10]

$$H^i(l) = L_M[J_s(l)] - \frac{Z_s}{\eta_0^2} L_E[J_s(l)], \tag{17}$$

Similarly, with a lossy dielectric surface the EFIE becomes

$$E^i(l) = L_E[J_s(l)] - Z_s L_E[J_s(l)]. \tag{18}$$

Since equations (17) and (18) each include only the unknown surface current J_s (and not M_s) they are well suited to solution using moment method techniques.

The hybrid MM/GTD technique can be extended to apply to equations (17) and (18) to find the scattering from lossy dielectric surfaces of the type shown in Figure 1 with little modification. The surface current between points A and D is again divided into pulse basis functions as described in equation (9), and the diffraction-current basis functions are unchanged from equation (7) since the diffracted field is still ray optical at suitable distances from the diffraction point [11]. The physical optics current does need to be modified slightly

BISTATIC SCATTERING FROM DIELECTIC CYLINDER
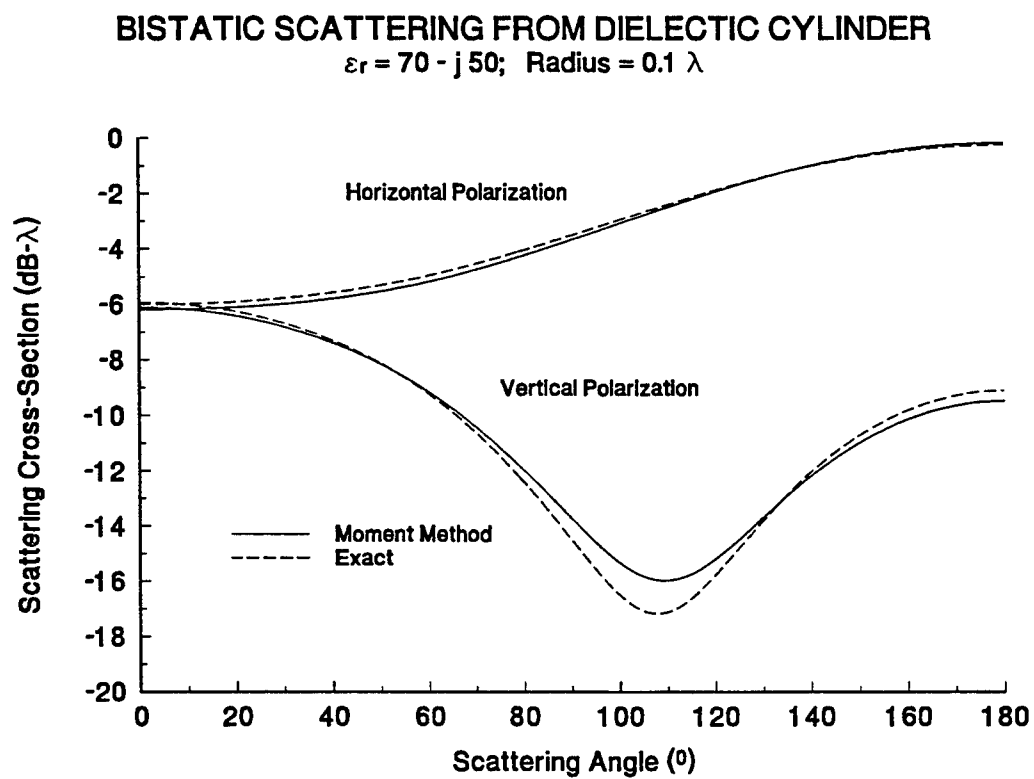$\varepsilon_r = 70 - j\,50; \quad \text{Radius} = 0.1\ \lambda$

Figure 4: Comparison of moment method and exact calculation of scattering from a lossy dielectric cylinder.
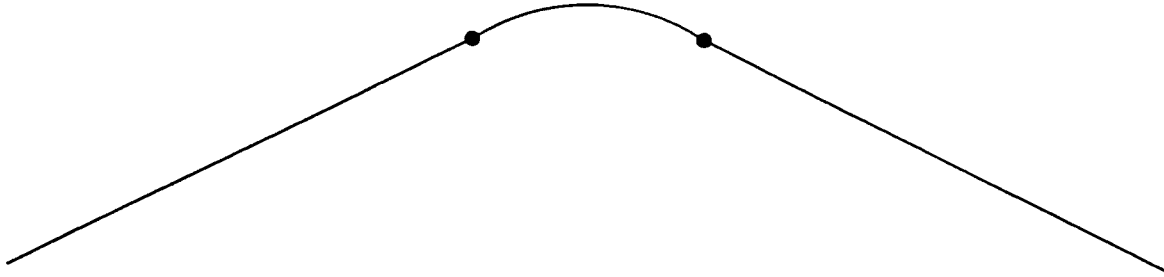
Figure 5: Rounded wedge scattering surface.

polarization (the electric field is parallel to the cylinder axis) and vertical polarization (the electric field is perpendicular to the axis), confirming the validity of the approach at both polarizations.

The scattering from a wedge with a rounded apex, as shown in Figure 5, was calculated using the full implementation of the hybrid MM/GTD technique. The radius of curvature of the apex was set at 0.5λ and the interior angle of the wedge was $120°$. The scattering was calculated with the real part of the dielectric constant fixed at 70 and imaginary part ranging from infinity (perfectly conducting case) down to 10. The results are shown in Figures 6 and 7. The incidence angles in the figures are referenced to vertical. At incidence angles ranging from $-30°$ to $30°$ the backscattering is dominated by specular reflection from the apex. (The singularities occuring at $-30°$ and $30°$ are due to specular reflection from the infinitely long extensions). Since the conditions of equation (15) are met, at these incidence angles the reduction in the backscattered field with decreasing surface conductivity should be directly proportional to the normal incidence, flat-surface reflection coefficient. When $\epsilon_r = 70 - j10$ the magnitude of the reflection coefficient is 0.788, indicating that the scattering from this surface should be very close to 2.1 dB below that from the perfectly conducting surface in this region. The actual reductions at $0°$ incidence agree with this value to within 0.15 dB at both polarizations.

At incidence angles beyond $30°$ $(-30°)$ the scattering is no longer dominated by specular reflection, but instead is due entirely to back diffraction, both from the discontinuity in the

[4] J. C. West, "Effect of shadowing on electromagnetic scattering from rough ocean-wave-like surface at small grazing angles", *IEEE Transactions on Geoscience and Remote Sensing*, 1995, under review.

[5] J. C. West, M. A. Sletten, and J. M. Sturm, "Comparison of numerically predicted scattering from a breaking wave with experiment", in *Proceeding of the Progress in Electromagnetic Research Symposium*, Seattle, Washington, 1995, p. 759.

[6] W. D. Burnside, C. L. Yu, and R. J. Marhefka, "A technique to combine the geometrical theory of diffraction and the moment method", *IEEE Transactions on Antennas and Propagation*, vol. AP-23, no. 4, pp. 551–558, July 1975.

[7] C. A. Balanis, *Advanced Engineering Electromagnetics*, Wiley, New York, 1989.

[8] T. B. A. Senior and J. L. Volakis, "Generalized impedance boundary conditions in scattering", *Proceedings of the IEEE*, vol. 79, no. 10, pp. 1413–1420, Oct. 1991.

[9] A. W. Glisson, "Electromagnetic scattering by arbitrary shaped surfaces with impedance boundary conditions", *Radio Science*, vol. 27, no. 6, Nov. 1992.

[10] W. V. T. Rusch and R. P. Pogorzelski, "A mixed-field solution for scattering from composite bodies", *IEEE Transactions on Antennas and Propagation*, vol. AP-34, no. 7, 1986.

[11] R. Tiberio, G. Pelosi, G. Manara, and P. H. Pathak, "High-frequnency scattering from a wedge with impedance faces illuminated by a line source, part i: Diffraction", *IEEE Transactions on Antennas and Propagation*, vol. 37, no. 2, Feb. 1989.

[12] H. T. Thacher, "Algorithm 215: SHANKS", *Communications of the ACM*, vol. 6, no. 11, pp. 662, Nov. 1963.

[13] W. H. Press, S. A. Teulkolsky, W. T. Wetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Programming*, Cambridge University Press, Cambridge, 2 edition, 1992.

[14] Jr. C. A. Siller, "Evaluation of the radiation integral in terms of end-point contributions", *IEEE Transactions on Antennas and Propagation*, vol. AP-23, no. 9, Sept. 1975.

[15] G. L. James, G. Tong, and D. A. Ross, "Uniform diffraction solution for a discontinuity in curvature", *Electronics Letters*, vol. 11, no. 23, pp. 557–559, Nov. 1975.

[16] P. H. Pathak, "An asymptotic analysis of the scattering of plane waves by a smooth convex cylinder", *Radio Science*, vol. 14, no. 3, pp. 419–435, May 1979.

ATM DS-3 EXPERIMENTS VIA THE ADVANCED COMMUNICATION TECHNOLOGY SATELLITE

Valentine Aalo and Okechukwu Ugweje

Electrical Engineering Department

Florida Atlantic University

500 North West 20th Street

Boca Raton, FL 33431

Mostafa Chinichian

Electrical Engineering Department

California Polytechnic State University

San Luis Obispo, CA 93407

14-1

ATM DS-3 EXPERIMENTS VIA THE ADVANCED COMMUNICATION TECHNOLOGY SATELLITE

Valentine Aalo
Electrical Engineering Department
Florida Atlantic University

Okechukwu Ugweje
Electrical Engineering Department
Florida Atlantic University

Mostafa Chinichian
Electrical Engineering Department
California Polytechnic State University

Abstract

A series of experiments were conducted via the ACTS DS-3 link carrying ATM traffic in a PLCP frame format. These experiments were performed between Rome Laboratory, Griffiss Air Force Base, New York and Communication Research Centre, Nepean, Canada. These tests were conducted to test the viability of ATM DS-3 bearers via the satellite at a performance level comparable to that of terrestrial fiber optic link or microwave link. In these tests, both single and multiple channels configurations of ATM satellite bearers were tested. The single channel experiments were configure to fully load the channel at 96,000 Cells per second, while the multiple channel experiments were used to assess the ability of ATM protocol to carry information from different sources at different rates. Our main objective of characterizing the DS-3 satellite channel for the transmission of ATM signals was achieved, and the result obtained were presented and analyzed.

ATM DS-3 EXPERIMENTS VIA THE ADVANCED COMMUNICATION TECHNOLOGY SATELLITE

Valentine Aalo

Okechukwu Ugweje

Mostafa Chinichian

1.0 INTRODUCTION

The popularity of Asynchronous Transfer Mode (ATM) has risen to a new height, and all efforts are being made to experimentally verify its potential and usefulness. It is indicated in [1], that in the next decade, most telecommunication traffic will be carried by ATM technology. ATM is now regarded as the communication protocol of the future and will play a significant role in the "information superhighway". The usefulness and adaptability of ATM to current technology is also the subject of lots of research in both Government and industry. ATM technology and its applications are still being developed and tested. Global deployment of ATM technology may take some time as most economical and effective means of implementing this evolving technology is still being studied.

Although ATM was originally intended for fiber optic links transmitting high speed data, it can be used for other links such as the satellite link, which is the subject of this experiment. Satellites will play a vital role in the provision of ATM-based services such as voice, data, video and imaging, on demand, at very high data rates and at any location. Many performance issues regarding the transfer of ATM information bearers via the satellite channel remain unresolved, especially at Ka-band where the performance of satellite communication systems are severely impaired by atmospheric propagation effects, especially rain attenuation. We are interested in studying by means of actual experiment those characteristics of ATM protocol that is suitable and reliable for the transmission of information via satellite Ka-band at DS-3 rates.

The aim of this test was to demonstrate the practicality of transmitting and receiving ATM Cells using the Advanced Communication Technology Satellite (ACTS). This report documents, in a summarized form, the experiment of transmitting ATM signal via the ACTS conducted in the Summer of 1995 at Rome Laboratory (RL), Griffiss Air Force Base (AFB), New York. These experiments are part of the global grid Technical Technology Cooperation Program (TTCP) network, in which RL will communicate via the satellite with the Canadians at the Communication Research Center (CRC) [2]. These experiments were conducted via the satellite between the

CRC, Nepean, Canada and RL, Griffiss Air Force Base (GAFB), New York. This experiment is part of the on-going effort to link the assets of the Air Force at RL, the Army at the Communication Electronic Command (CECOM), and the Navy at the Naval Research and Development (NRaD) [3]. This report is based on the segment of the experiments that were conducted from 27 June 1995 through 24 August 1995. Several tests were performed via a satellite DS-3 link carrying ATM traffic in a Physical Layer Convergence Protocol (PLCP) frame format. In these tests, both single and multiple channels configuration of ATM satellite bearers were tested. The single channel configuration fully loads the channel at 96,000 Cells/s, while the multiple channel configuration was used to assess the ability of ATM protocol to carry information from different sources at different transmission rates.

2.0 SETUP AND CONFIGURATIONS

The first stage of this experiment involved the characterization of the satellite channel for reliable transmission of ATM signals at DS-3 rates via the ACTS. In the second stage, ATM Cells bearers were transmitted and received via the ACTS satellite at DS-3 rates. In the following sections the resources and experimental procedure used in these tests, particularly at the RL test site, are briefly described.

2.1 Test Sites and Facilities

As indicated earlier, as part of the global grid network, RL, New York, will communicate via the satellite with CRC, Canada. The basic configuration used to connect CRC and RL via the ACTS is shown in Figure 1. This configuration simply shows the terminal equipments and the ATM equipments in block diagram form. Details are not shown. This configuration includes the ATM generator/analyzer equipment and a PC workstation fitted with ATM adapter card. This setup is briefly described below.

2.1.1 Rome Laboratory, Griffiss AFB, New York

At RL, the satellite transmission facilities is located outside in two huts south of building 3. The high data rate antenna is 1.8m with gain of 60.40 dBi for Uplink and 48.53 dBi for Downlink. The antenna is positioned at

> Latitude: 43° 13', 13.18" N
> Longitude: 75° 24', 34.00" W
> Azimuth 214.1°; Elevation 34.68°; Polarization Tilt 24.43°

In addition to the 1.8m high data rate antenna, other terminal equipments at the RL test site consist of the EF
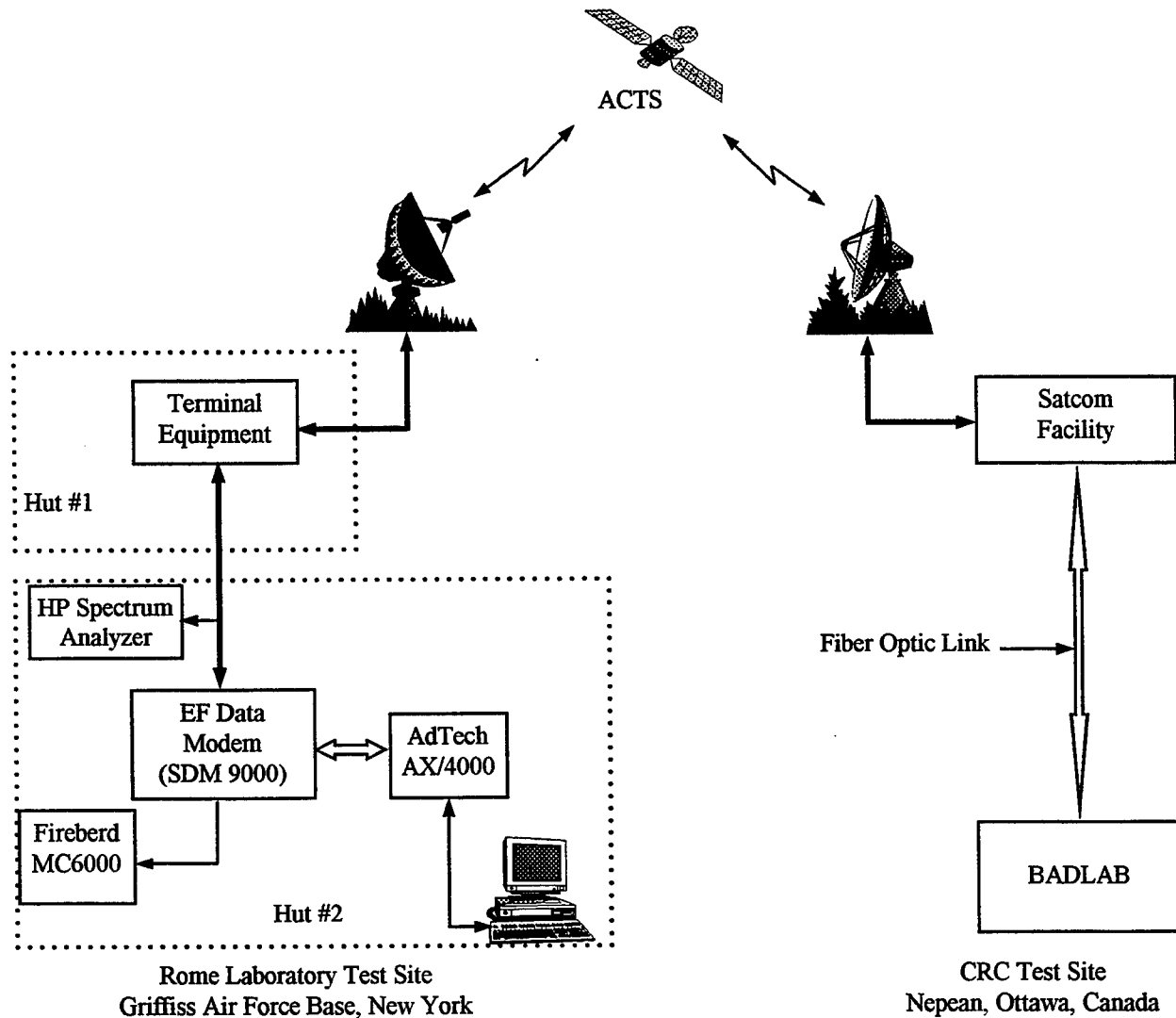
Figure 1: ATM DS-3 Test Configuration

Data modem (model SDM-9000), Ka-band traveling-wave-tube amplifier (TWTA), test loop translator (TLT), converters and amplifiers. The EF Data modem is a high performance, full-duplex, digital-vector modem with data rate capabilities of 6 Mbps to 51.84 Mbps. The modem has built-in scrambler/descrambler, differential encoding/decoding, multi-rate forward error correction capabilities (convolutional encoder and Viterbi decoder) and can be configured to add overhead/framing to the data. For example, a Reed-Solomon (RS) encoding and decoding is provided to work with the built-in Viterbi decoder, in conjunction with additional framing and interleaving, resulting in improved overall performance.

At the RL test site, the ATM generator/analyzer, AX/4000, manufacture by AdTech, Inc. is used as the ATM

test-bed equipment. This equipment provides complete signally, switching, generation and analysis of ATM traffic. It is modular in nature, allowing for custom configuration for a variety of ATM test applications. This unit consists of the mainframe, the ATM generation/analyzer modules and is connected to a window-based workstation. While a variety of port interfaces are possible (for example, SONET OC-3c, E3, TAXI, and DS-3), the interface provided with the AX/4000 is the DS-3 interface using electrical format and BNC connection to the EF Data modem, and IEEE 804.2 interface to the PC workstation. Using the associated AX/4000 Microsoft window-based software, one can select different configurations of single or multiple ports (each port has up to 16 channels or substreams) with capabilities of up to 140 Mbps. However, in this experiment, we are only interested in the DS-3 speed of 45 Mbps, and the AX/4000 is capable of generating a full DS-3 Cell stream.

The AX/4000 ATM equipment is installed in the adjacent hut, about 25 feet from the first hut as shown in Figure 1. The Fireberd Communication Analyzer (model MC6000), HP Spectrum Analyzers (HP 8568A & HP 8563), and the EF Data modem are co-located with the ATM equipment. Note that all connections at the RL test site uses electrical format.

2.1.2 Communication Research Centre, Nepean, Canada

The ATM test-bed at the CRC test site is located in the Broadband Application and Demonstration Lab (BADLAB) in building 2D [4]. The satellite communication (satcom) facility is located in building 46. The satcom facility and the ATM facility are linked by fiber optic line, over off-net extension via the satellite ATM links.

The antenna at CRC test site is 4.2m with gain of 52.34 dBi for Uplink and 56.59 dBi for Downlink. The position of the antenna is at

Latitude: $45° 21'$, 9.95" N
Longitude: $75° 54'$, 9.59" W
Azimuth $220.85°$; Elevation $29.16°$; Polarization Tilt $48°$

The terminal facilities (Modem, TLT, TWTA) at CRC test site is the same as RL test site. There may be minor variations in terms of the location and connectivity of the equipments but the functionality of all the equipments at the two test sites remain the same. It is perhaps important to note that most of the terminal equipments used in this experiment at both test sites were provided by Canada's CRC in corporation with Air Force's Rome Laboratory. Their contribution include the frequency plan, the redesign of the converters and amplifiers to be compatible with ACTS Ka-band frequencies at DS-3 rates, and the actual test procedure used in these experiments.

2.2 Satellite Facilities

The Advanced Communication Technology Satellite (ACTS) was used for this experiment. Access to the ACTS was provided by NASA Lewis, Cleveland, Ohio. The ACTS operations center at NASA Lewis provided all access for the East Scan 4E spot beam used by RL test site as well as the Steerable spot beam used by CRC.

During these tests, RL transmitted horizontally with polarization tilt of 24.4° clockwise into the East Scan 4E beam at 29.125 GHz and received in the orthogonal plane at 19.505 GHz. On the other hand, CRC, transmitted vertically with polarization tilt of 22.2° clockwise into the steerable antenna at 29.225 GHz and received orthogonally at 19.405 GHz [5]. Because of the enormous loss in orthogonal polarization as seen from the satellite EIRP and G/T coverage, self loopbacks are not practical. It should be pointed out, however, that the ACTS operation center at NASA Lewis can provide satellite loopbacks for both the steerable and East Scan 4E spot beams. This implies that NASA Lewis can serve as a loopback node for either RL test site or CRC test site.

3.0 EXPERIMENTAL PROCEDURE

3.1 Channel Characterization

The sequence of tests involving the transmission of ATM Cells bearers via the ACTS started on 27 June 1995. The first set of tests were aimed at characterizing the channel between RL, and CRC, and to familiarize RL personnel with the ground terminal equipment. The objective of the channel characterization experiments is to evaluate the performance of the EF Data modem with the Fireberd tester at DS-3 rates over the ACTS. For the channel characterization experiments, the setup is as shown in Figure 1 without the AdTech AX/4000 equipment connected.

Two types of modulation schemes, namely, the QPSK and 8PSK were emphasized in this experiment. The characterization tests as well as the ATM transmission tests were conducted using both QPSK and 8PSK with and without RS encoding for different modem power levels. During this period of characterization, a number of transmissions involving continuous wave (CW) at the two modulation schemes were performed. This was done to evaluate the EF Data modem and the Fireberd DS-3 test equipment. The channel characterization and modem evaluation are necessary to ensure that at DS-3 rates and in the Ka-band the modem specifications for the different modulations can be verified. During the CW transmissions, the transmit power levels at both ends (CRC and RL) were varied (with the TX power meter reading recorded) while the carrier power (C) and the noise power (N_o) are read from the Spectrum Analyzer at opposite ends.

The bit error rate (BER) is measured using the Fireberd Analyzer. It is also possible to measure the BER using the EF Data modem but we did not use the measurement from the modem. To evaluate the modem and the Fireberd BER tester, the QPSK with RS, QPSK without RS, 8PSK with RS and 8PSK without RS were used as the modulation schemes. QPSK modulation is at the code rate of 3/4 while 8PSK is at the code rate of 2/3. In each case, both the received BER and E_b/N_o were measured for each setting of the transmit power. All transmissions, by default, included the modem's built-in convolutional and Viterbi decoding at the indicated code rates.

At each transmit modem power level, the transmit power meter reading were recorded, as well as the received C and N_o at the other end. From these measurements, the E_b/N_o is given by $E_b/N_o = C/N_o - 10 \log(44.73 \times 10^6)$, where C and N_o are measured with the Spectrum Analyzer

3.2 ATM Transmission Experiments

A series of experiments were conducted via the ACTS DS-3 link carrying ATM traffic in a PLCP frame format. The main aim of the ATM experiments is to characterize the DS-3 satellite link for the transmission of ATM signals. Specifically, some ATM QoS parameters were measured using different levels of modulation (QPSK, 8PSK) with forward error correction and V.35 data scrambling. For the ATM experiments, the setup is as shown in Figure 1, with the AdTech equipment connected. As indicated earlier, the AdTech equipment , AX/4000, is used as the ATM test-bed. The desired configuration is set by means of the window-based software accompanying the AX/4000. There are two types of transmission configurations (single and multiple channels) used in these tests. In the single channel experiment, one port and one substream was selected and configured with the following parameters:

Substream 1
- AAL Type: Test Cells
  PRBS Type: $2^9 - 1$
- GFC Range: 0h - fh
  VPI Range: 00h - ffh
  VCI Range: 0010h
  PT Range: 0 - 3
  CLP Range: 0 - 1
- Cell Rate: 96,000 cells/s

The single channel payload data stream was set to generate 96,000 cells per second. This is the maximum cell generation capacity of the AX/4000. All other ports and substreams (substream 2 to 16) are disabled.

For the multiple channel case, one port and six substreams (substream 1 to 6) are selected and configured as shown in Table 1. Substreams 7 to 16 are disabled. Table 1 shows that substreams 1 & 2, substreams 3 & 4,

Table 1: Multiple Channel Configuration PORT 1 SETUP					
Substream 1 AAL Type: Test Cells PRBS Type: 2^9-1	Substream 2 AAL Type: Test Cells PRBS Type: 2^9-1	Substream 3 AAL Type: Test Cells PRBS Type: 2^9-1	Substream 4 AAL Type: Test Cells PRBS Type: 2^9-1	Substream 5 AAL Type: Test Cells PRBS Type: 2^9-1	Substream 6 AAL Type: Test Cells PRBS Type: 2^9-1
GFC Range: 0h-fh VPI Range: 00h-ffh VCI Range: 0001h PT Range: 0 - 7 CLP Range: 0 - 1	GFC Range: 0h-fh VPI Range: 00h-ffh VCI Range: 0002h PT Range: 0 - 7 CLP Range: 0 - 1	GFC Range: 0h-fh VPI Range: 00h-ffh VCI Range: 0003h PT Range: 0 - 7 CLP Range: 0 - 1	GFC Range: 0h-fh VPI Range: 00h-ffh VCI Range: 0004h PT Range: 0 - 7 CLP Range: 0 - 1	GFC Range: 0h-fh VPI Range: 00h-ffh VCI Range: 0005h PT Range: 0 - 7 CLP Range: 0 - 1	GFC Range: 0h-fh VPI Range: 00h-ffh VCI Range: 0006h PT Range: 0 - 7 CLP Range: 0 - 1
Cell Rate: 150 Cells/s	Cell Rate: 150 Cells/s	Cell Rate: 604 Cells/s	Cell Rate: 604 Cells/s	Cell Rate: 3640 Cells/s	Cell Rate: 3640 Cells/s

substreams 5 & 6, have ATM Cell generation rates of 150 Cells/s, 604 Cells/s, and 3640 Cells/s respectively. The combined Cell Rate should be less or equal to 96,000 Cells/s. In our own case, for multiple channel configuration, the combined Cell Rate is 8788 Cells/s. In either the single substream or multiple substreams configuration, the PRBS errors are triggered by the Cell Sequence Error, Cell Payload Bit Error or the Loss of Payload Pattern Synchronization.

Using the AdTech AX/4000, it is possible to measure a lot of information for each test transmission. These are known as stream and substream statistics. A sample of the measured statistics using the AX/4000 analyzer module is shown in Table 2. This table shows a sample of the statistics for QPSK experiment conducted on 24 July 1995, and the 8PSK experiment conducted on 11 August 1995. Both experiments are for multiple channel (or substream) tests, with only the first substream shown. Similar statistics was obtained for the single channel tests.

4.0   ANALYSIS OF RESULT

The key issues regarding the reliable transmission of ATM signal via satellite is the subject of this test, especially, the error characteristic and propagation delay. This is because in satellite transmission at the Ka-band, the satellite link is susceptible to atmospheric effects, the most important of which is attenuation due to rain. To this effect, the weather condition at RL during each period of an experiment is recorded and is enclosed in the appendix. The reason for this is to observe any relationship between the collected data and the weather condition. We observed that on several conditions, the satellite link between RL and CRC was lost whenever there is a rain-

Table 2: ATM Transmission Stream and Substream Statistics	
QPSK Modulation	**8PSK Modulation**

Stream Statistics:
    Aggregate Count: 48343
    Uncorrected Header Count: 915
    Aggregate Cell Rate: 70,113 cells/s
    Aggregate Cell Transfer Capacity: 29.728 Mb/s
    Bandwidth Percentage: 73.03 %
	Stream Statistics:
	    Aggregate Count: 17114
	    Uncorrected Header Count: 3
	    Aggregate Cell Rate: 22,240 cells/s
	    Aggregate Cell Transfer Capacity: 9.430 Mb/s
	    Bandwidth Percentage: 23.17 %

Substream Statistics:
    Cell Count: 177695 cells
    Cell Rate: 149.8 cells/s
    Cell Transfer Capacity: 0.064 cells/s
    CLP=1 Cell Count: 41 cells
    CLP=1 Error Ratio: 2.31E-04
    Cell Loss Count: 124 cells
    Cell Loss Ratio: 0.000697495
    Misinsertion Cell Count: 40 cells
    Cell Misinsertion Rate: 0.0337286 cells/s
    Out-of-Sequence Count: 0 events
    Errored Cell Count: 85 cells
    Cell Error Ratio: 0.000478348
    PRBS Bit Error Count: 895 bits
    PRBS Bit Error Rate: 1.65682e-05 errors/bit
    PRBS Sync Error Count: 0 resyncs
	Substream Statistics:
	    Cell Count: 556180 cells
	    Cell Rate: 150.0 cells/s
	    Cell Transfer Capacity: 0.064 cells/s
	    CLP=1 Cell Count: 0 cells
	    CLP=1 Error Ratio: 0.00E+00
	    Cell Loss Count: 9 cells
	    Cell Loss Ratio: 1.61815e-05
	    Misinsertion Cell Count: 0 cells
	    Cell Misinsertion Rate: 0 cells/s
	    Out-of-Sequence Count: 0 events
	    Errored Cell Count: 93 cells
	    Cell Error Ratio: 0.000167212
	    PRBS Bit Error Count: 215 bits
	    PRBS Bit Error Rate: 1.2716e-06 errors/bit
	    PRBS Sync Error Count: 0 resyncs

storm in RL, CRC or in between. This observation is consistent with theoretical analysis which has shown that rain attenuation severely affects transmission at Ka-bands.

To properly assess the viability of ATM via satellite, the cumulative effect of the satellite link characteristics on the ATM parameters must be determined. Performance consideration for ATM are currently based on the assumption that transmission bit errors are randomly distributed. This may be true, to a high degree of accuracy for most terrestrial microwave and fiber optic based transmission systems. However, the validity of this assumption has not been verified for satellite ATM Cell bearers. The bursty nature of transmission errors via the satellite requires careful evaluation.

4.1 Channel Characterization

For satellite transmission channels, ATM performance parameters can only be quantified if the nature of the transmission bit errors is first characterized. The characterization phase of this experiment was aimed at achiev-

ing this objective.

The entire raw data collected during this experiment can be obtained from the satcom division, RL/C3BA, Rome Laboratory, New York. Some of these data is presented here for the purposes of discussion.

Most of the data presented here are collected from the RL test site. The other half collected from CRC can also be obtained from RL/C3BA or from CRC. For brevity, this other half is not presented here.

The graph of E_b/N_o versus BER for the two modulation schemes, with or without RS encoding is shown in Figures 2 and 3. In Figure 4 and 5, the performance of QPSK and 8PSK are compared. The effect of using RS encoding is shown in Figure 6, while the characteristics of both the forward (RL to CRC) and backward (CRC to RL) links are shown in Figure 7. The actual experiment represented by each graph is also indicated on the plot. The following observation can be made from these plots.

- QPSK provides better performance than 8PSK. This result could be attributed to the fact that the EF Data modem is more stable with respect to QPSK than the 8PSK. In other words, it is much simpler for the EF Data modem to distinguish between four phases as opposed to eight phases. Thus, signal detection is more reliable at QPSK modulation than at 8PSK modulation.

- RS encoding/decoding provides considerable system performance improvement. For example in QPSK at BER = 10^{-6}, there is a power saving of as much as 1.5 dB with the use of RS coding. The RS coding gain increases for lower BER values. This is consistent with expected analytical values.

- From Figure 7, it is observed that RL to CRC link and the reverse link have identical transmission link characteristics. The noticeable variation could be attributed to different antenna gain at both test sites, and to the different ACTS transponders (or spot beams) used for both test sites.

4.2 ATM Transmission

The bursty nature of the error statistics and the corresponding large propagation delay involved in a satellite link affect ATM transmission performance [6]. A number of ATM parameters have been identified as being very important in the assessment of the performance of an ATM network. Some of the most important QoS parameters include the transmission channel bit error ratio (BER), cell loss ratio (CLR), errored cell count (ECC) and the PRBS parameters. In the experiments, emphasis was placed on those parameters and statistics that can be measured by the AdTech ATM test equipment. Some of these parameter is discussed bellow.

- *Bit Error Rate (BER):* This is the rate at which the transmitted bits were changed in the ATM physical layer.

- *Cell Loss Ratio (CLR):* The ratio of the number of lost ATM cells sent by a user in specified time interval. Due to the random nature of the ATM Cells, the limited size of the ATM traffic and the limited size of the buffers, it is usually possible that a cell arriving at a switching node may be lost . Thus CLR are caused by buffer overflows and bit error in the cell header that can be detected but not be corrected.

- *Cell Misinsertion Ratio (CMR):* Defined as the ratio of the cells delivered to a wrong destination to the total number of cells sent. It occurs as a result of an undetected error in the header that causes a change of the cell destination.

4.2.1 Single Channel

Table A1 in the appendix shows a sample of raw data collected for single channel ATM transmission. While many signal channel runs were made, a selected number of them are plotted in Figure 8 to 13 to illustrate the results obtained. The particular experiment and the date performed are indicated on the plots. From these Figures the following observations may be made:

- CLR and CER appear to have similar relations with channel BER. They both decrease as BER decreases.
- CLC and ECC are lowest at BER $=10^{-5}$ for the range of BER $=10^{-6}$ to 10^{-4}.
- For the received ATM alarms, it is observed that a) Framing rate appears to take its highest value at BER $= 10^{-5}$, and takes its minimum value in the neighborhood of 10^{-6}, and b) FEBE and P-Bit rate appear to increase steadily with increasing BER.
- For the PLCP alarms, the FEBE rate and P-Bit rate appear to have similar relations with channel BER.
- The forward and backward satellite channels between RL and CRC appear to be symmetric with identical measured ATM parameters.

4.2.2 Multiple Channel

A sample of the raw data collected for the multiple channel ATM transmission is shown as Table A2 in the appendix. As many as six ATM substreams of varying cell rates were combined in the multiple channel case with cell generating rate of 8788 Cells/s. As shown in Table A2, two types of measured parameters may be distinguished. One set has to do with he alarms that are generated for the composite stream, mainly, received frame alarm and received PLCP alarm. The other set of are the ATM Cell characteristics of each of the individual substreams as shown in Table 2. Figures 14 to 16 show the multichannel ATM parameters as function of channel BER. The following observations may be made regarding the Figures.

- Substreams with similar configurations exhibit similar characteristic with respect to the ATM QoS pa-

rameters.

- The received ATM alarms, FEBE, P-Bit, and Framing rates exhibit similar functional relationship with the channel BER.
- The effect of RS coding on the ATM transmission was not evident. However, it is conjectured that the use of RS coding will improve performance considerably.

6.0 CONCLUSION

In this experiment, we have successfully transmitted and received ATM Cells between Rome Laboratory, GAFB, New York and CRC, Canada. These ATM tests were performed at Ka-band frequencies over the ACTS. Two transponders (or beams) - the East Scan 4E spot beams and the Steerable spot beam, were used and the signal modulation schemes employed were the QPSK and 8PSK. It is observed in our experiment that the modulation scheme used affects the stability of the received signal. Also, during these experiments, we observed that rain attenuation has a dramatic effect in satellite transmissions at Ka-band frequencies. Ways of mitigating this effect should be the focus of more studies.

Although we were able to transmit and receive ATM Cells at Ka-band, it will be premature to draw a conclusion on the performance of ATM transmission over satellite as a result of one set of experiments. It is strongly recommended that more experiments be conducted after which a more meaningful conclusion on the performance of ATM signals over satellite can be reached. Some, if not all, of the test performed in this experiment should be repeated in other to verify the validity of these test results.

REFERENCES

[1]   Arthur Miller, "From here to ATM - (Special Report)", IEEE Spectrum, June 1994.

[2]   G. A. Bivens, "Satellite Networking Research in Scaleable Networking Technology, Rome Laboratory Technology Demonstration with NASA ACTS Satellite - Proposal", May 1994.

[3]   V. Aalo and O. Ugweje, "A Program Plan for Transmitting High-Data -Rate ATM/SONET Signals over the ACTS, Final Report AFOSR, September 1994.

[4]   J. Butterworth and G. Nourry, "TEST PLAN - for an investigation of THE TRANSPORT OF ATM-BASED TRAFFIC BY Ku-BAND SATELLITE BEARERS, Workshop on the "Global Grid" concept", Sponsored by TTCP STP-6, RL/CRC Internal Document, 16th January 1995.

[5]   Corey Pike, Rome Lab/CRC internal document

[6]   R.O. Onvural, Asynchronous Transfer Mode Networks Performance Issues, Boston; Artech House, 1994.
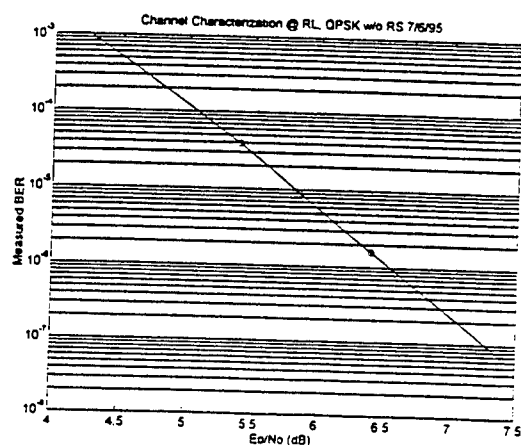
Figure 2. QPSK Channel Characterization without
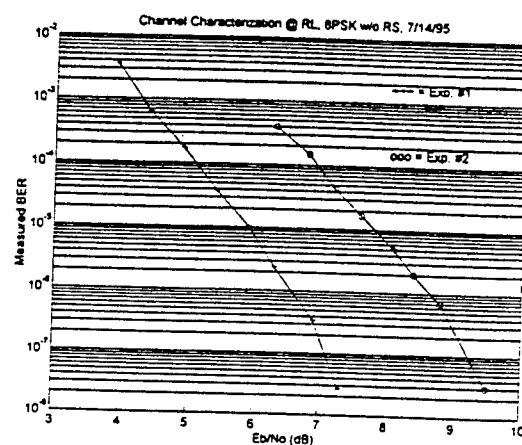Reed Solomon Encoding

Figure 3. 8PSK Channel Characterization without
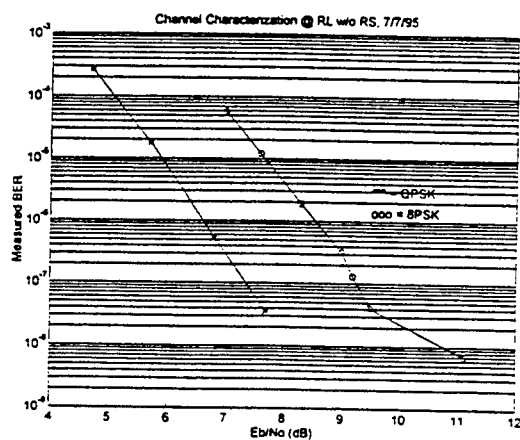Reed Solomon Encoding

Figure 4. QPSK and 8PSK Channel Characterization
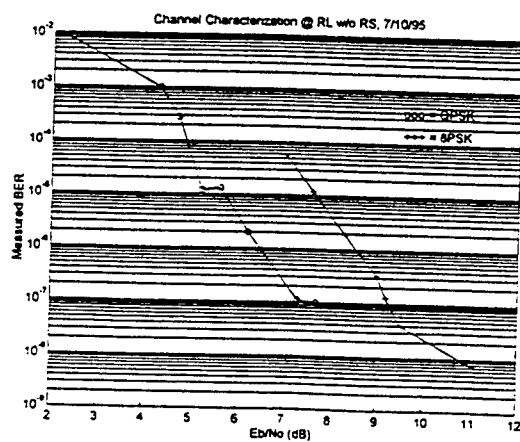without Reed Solomon Encoding

Figure 5. QPSK and 8PSK Channel Characterization
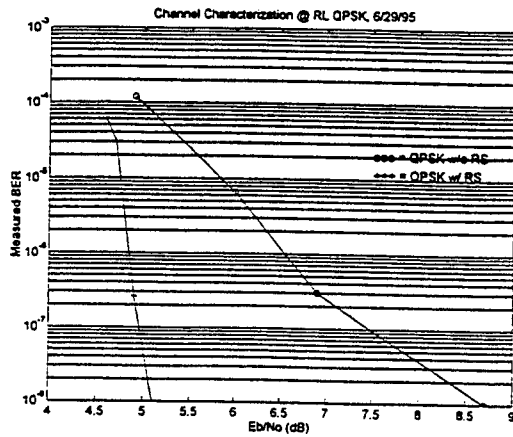without Reed Solomon Encoding

14-14

Figure 6. The Effect of Reed Solomon Encoding on Channel Characterization
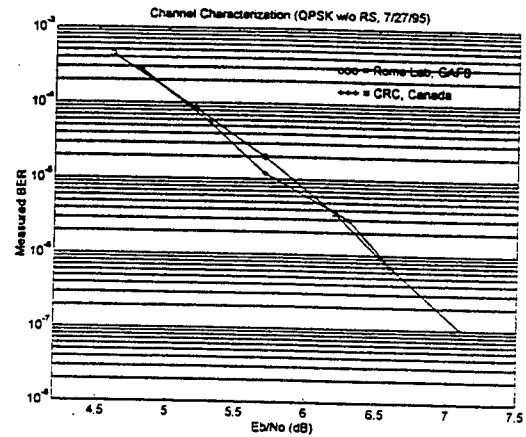

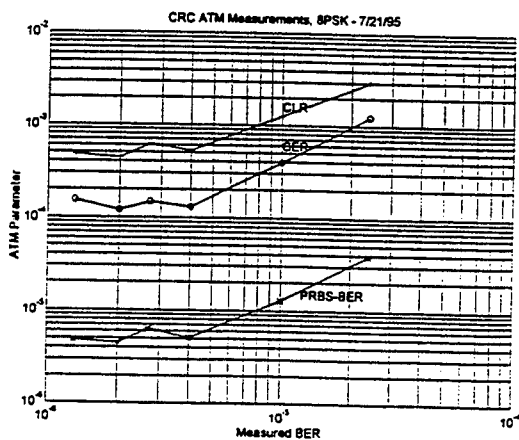Figure 7. Forward and Backward Link Characteristics w/o Reed Solomon Encoding

Figure 8. ATM Measurements - 8PSK

Figure 9. CLC & ECC Measurements - 8PSK
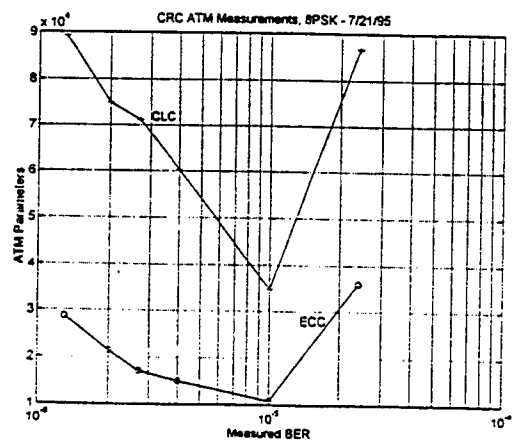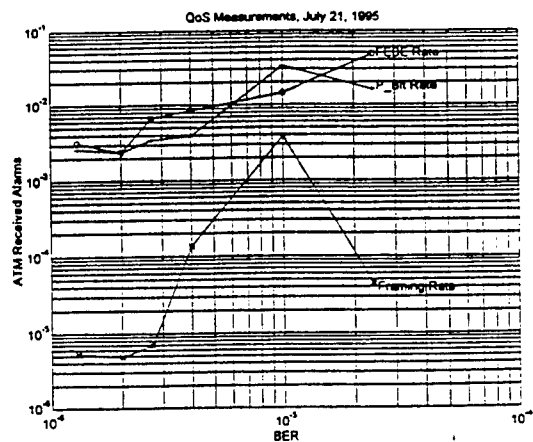
14-15

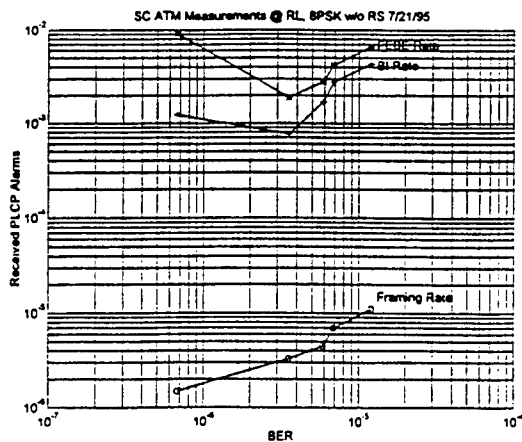Figure 10. Rx Frame Alarms - 8PSK
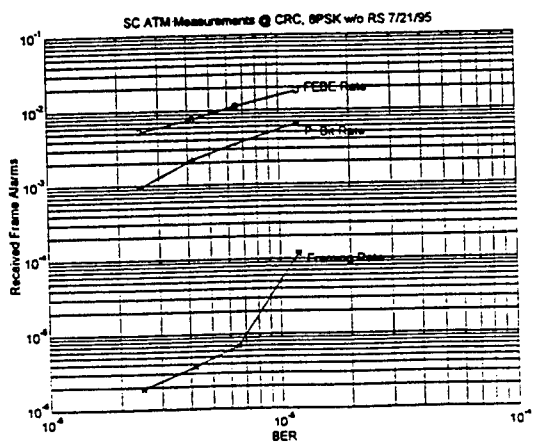

Figure 11. Rx Frame Alarms - QPSK

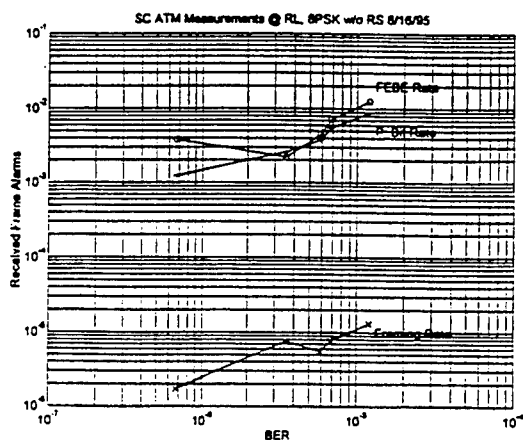Figure 12. Single Channel w/o RS Measurements
@ CRC - 8PSK

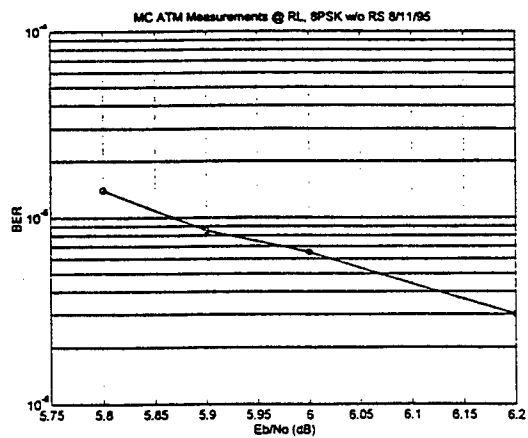Figure 13. Single Channel w/o RS Measurements
@ RL - 8PSK

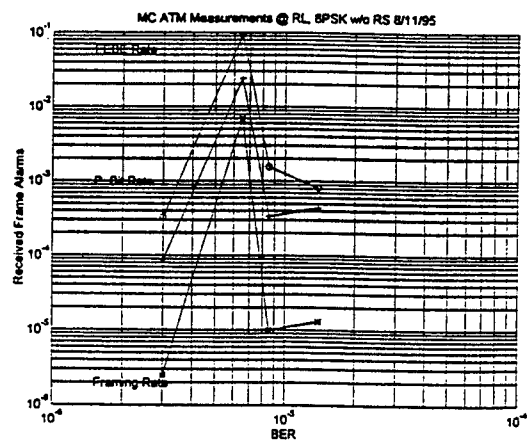14-16

Figure 14. Multiple Channel w/o RS
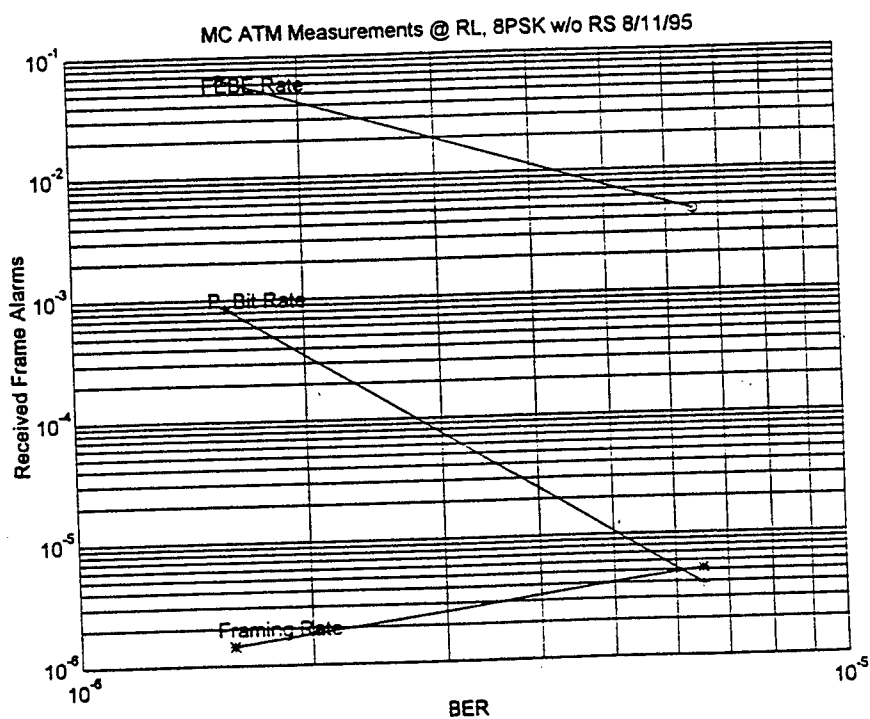@ RL - 8PSK

Figure 15. Multiple Channel Alarms w/o RS
@ RL - 8PSK

Figure 16. Multiple Channel Alarms w/o RS @ RL - 8PSK

14-17

Appendix

A: Weather Conditions and Schedule of Experiments

#	Date	Time	Weather	Comments
1.	27 June 95 (Tuesday)	1815-1930 EDT	Sunny	Terminal Configuration/Setup
2.	29 June 95 (Thursday)	1230-1630 EDT	Sunny	Modem Configuration
3.	5 July 95 (Wednesday)	1200-1600 EDT	Hazy	Modem Configuration
4.	6 July 95 (Thursday)	0800-1100 EDT	Hazy & Hot	C/No Measurement
5.	7 July 95 (Friday)	1300-1545 EDT	Hazy & Hot	BER, Eb/No measurement (QPSK)
6.	10 July 95 (Monday)	1045-1430 EDT	Cloudy	CW measurement
7.	14 July 95 (Friday)	1030-1400 EDT	Clear	8-PSK/ADTECH
8.	17 July 95 (Monday)	1045-1430 EDT	Sunny	ATM (QPSK, No RS)
9.	21 July 95 (Friday)	0730-1230 EDT	Cloudy	ADTECH problem
10.	24 July 95 (Monday)	1045-1430 EDT	Cloudy	ADTECH problem (QPSk, No RS)
11.	27 July 95 (Thursday)	1430-1700 EDT	Cloudy	ADTECH problem (QPSk, No RS)
12.	3 Aug. 95 (Thursday)	1100-1500 EDT	Overcast	Modem problem - no data
13.	4 Aug. 95 (Friday)	1100-1500 EDT	Cloudy	Modem problem - no data
14.	7 Aug. 95 (Monday)	1115-1530 EDT	Sunny	RL loop-back test-Modem problem
15.	11 Aug. 95 (Friday)	0900-1300 EDT	Cloudy	Modem Configuration
16.	14 Aug. 95 (Monday)	0800-1200 EDT	Sunny	ATM (8-PSK, no RS, 6 substreams)
17.	16 Aug. 95(Wednesday)	0800-1100 EDT	Sunny	ATM (8-PSK, no RS, 6 substreams)
18.	20 Aug. 95 (Sunday)	1500-1930 EDT	Sunny	RL loop-back (ATM, QPSK)
19.	22 Aug. 95 (Tuesday)	0915-1300 EDT	Sunny	ATM (QPSK, No RS, 6 Streams)
20.	25 Aug. 95 (Friday)	0800-1100 EDT	Sunny	ATM (QPSK, No RS, 6 Streams)

Table 3. A Sample of the raw data collected for Single Channel ATM Test

Partial Results For Single Channel Test 21 July 95, 8PSK without Reed-Solomon Coding (CRC Blue Sky - Rome Overcast/Rain)

BER	Cell Count	CLC	MCC	OOS Count	ECC	Cell Rate	CLR	CMR	CER	PRBS BEC	PRBS BER	PRBS SEC
2.4E-5	2.88E+7	35,926	0	1	86,697	95,008	1.24E-3	0	3.01E-3	339,085	3.87E-5	0
1.0E-5	2.74E+7	10,822	0	0	34,678	90,654	3.94E-4	0	1.26E-3	106,500	1.28E-5	0
4.0E-6	1.15E+8	14,987	0	1	60,280	95,763	1.3E-4	0	5.24E-4	177,390	5.08E-6	0

BER: Bit Error Rate;   CLC: Cell Loss Count;   MCC: Misinsertion Cell Count;   OOS: Out of Sequence;
ECC: Errored Cell Count;   CLR: Cell Loss Ratio;   CMR: Cell Misinsertion Rate;   CER: Cell Error Ratio;
PRBS BEC: PRBS Bit Error Count;   PRBS BER: PRBS Bit Error Rate;   PRBS SEC: PRBS Synchronization Error Count

Rx Frame Alarm

BER	Line Code Count	Line Code Rate	Framing Count	Framing Rate	P-Bit Count	P-Bit Rate	CP-Bit Count	CP-Bit Rate	FEBE Count	FEBE Rate
2.4E-5	0	0	4,025	4.56E-5	91,049	1.6E-2	90,815	1.9E-2	136,481	4.79E-2
1.0E-5	0	0	336,710	3.81E-3	191,829	3.37E-2	191,761	2.24E-2	41,072	1.44E-2
4.0E-6	0	0	48,026	1.37E-4	87,800	3.89E-3	87,676	2.59E-2	95,994	8.51E-3

EF Data Modem Measurements

BER (Before)	Tx Power	Eb/No	BER (After)
2.4E-5	-10.8 dBm	7.4 dB	3.4E-5
1.0E-5	-10.3 dBm	7.7 dB	1.45E-5
4.0E-6	-10.1 dBm	8.0 dB	6.4E-6

Rx PLCP Alarm

BER	Framing Count	Framing Rate	B1 Count	B1 Rate	FEBE Count	FEBE Rate
2.4E-5	25,021	3.58E-5	265,534	1.37E-2	362,906	1.87E-2
1.0E-5	7,317	1.05E-5	103,740	5.35E-3	105,783	5.45E-3
4.0E-6	11,616	4.2E-6	179,242	2.33E-3	245,200	3.19E-3

Table 4. A Sample of the raw data collected for Multiple Channel ATM Test

Partial Results For Multiple Channel Test 22 August 95, QPSK without Reed-Solomon Coding (CRC clear Sky - Rome Sunny)

Stream #	BER	Cell Count	CLC	MCC	OOS Count	ECC	Cell Rate	CLR	CMR	CER	PRBS BEC	PRBS BER	PRBS SEC
1	8.53E-6	215,263	101	0	0	74	149	4.69E-4	0	3.44E-4	562	8.59E-6	0
2	8.53E-6	215,260	104	0	0	71	149	4.83E-4	0	3.30E-4	702	1.07E-5	0
3	8.53E-6	865,334	401	0	0	253	598	4.63E-4	0	2.92E-4	2,424	9.21E-6	0
4	8.53E-6	865,341	393	0	0	263	598	4.54E-4	0	3.04E-4	2,053	7.80E-6	0
5	8.53E-6	5,224,675	2,425	0	0	1,611	3,611	4.64E-4	0	3.08E-4	14,483	9.12E-6	0
6	8.53E-6	5,224,663	2,437	0	1	1,644	3,611	4.66E-4	0	3.15E-4	14,230	8.96E-6	0

BER: Bit Error Rate; CLC: Cell Loss Count; MCC: Misinsertion Cell Count; OOS: Out of Sequence;
ECC: Errored Cell Count; CLR: Cell Loss Ratio; CMR: Cell Misinsertion Rate; CER: Cell Error Ratio;
PRBS BEC: PRBS Bit Error Count; PRBS BER: PRBS Bit Error Rate; PRBS SEC: PRBS Synchronization Error Count

Rx Frame Alarm

BER	Line Code Count	Line Code Rate	Framing Count	Framing Rate	P-Bit Count	P-Bit Rate	CP-Bit Count	CP-Bit Rate	FEBE Count	FEBE Rate
8.53E-6	0	0	4,189	9.94E-6	9,055	3.33E-4	8,794	2.16E-4	20,961	1.54E-3

Rx PLCP Alarm

BER	Framing Count	Framing Rate	B1 Count	B1 Rate	FEBE Count	FEBE Rate
8.53E-6	28,961	8.89E-6	196,563	2.12E-3	319,966	3.46E-3

EF Data Modem Measurements

BER (Before)	Tx Power	Line Code Count	Eb/No	Framing Count	BER (After)
8.53E-6	-14.6 dBm	0	5.9 dB	4,189	1.37E-5